IMPROVE SSME POWER BALANCE MODEL

George C. Marshall Space Flight Center

and

The University of Alabama in Huntsville

FINAL REPORT

submitted to

NASA Headquarters
Washington, D.C.

Principal Investigator:

Dr. Gerald R. Karr, Chairman
Department of Mechanical and Aerospace Engineering
University of Alabama in Huntsville

NASA Contract NAS8-36955/D.O. 115

August 1992

(NASA-CR-184464) IMPROVE SSME POWER BALANCE MODEL Final Report (Alabama Univ.) 83 p

N93-10756

Unclas

G3/20 0118094

TABLE OF CONTENTS

1.0 BACKGROUND

As detailed in the original Scope of Work for this research effort, the principal investigator was to improve the steady-state power balance model (PBM) of the Space Shuttle Main Engine (SSME) in a three phase effort. A summary of the tasks in each phase is given below.

Phase 1: Construct software to facilitate SSME performance prediction and test data validation.

Phase 2: Review computational logic within the current version of the SSME PBM and implement programming structure.

Phase 3: Develop programming logic to improve the physical consistency of routing routines within the SSME PBM.

After discussions with John Butas of the NASA/MSFC Propulsion Laboratory in January of 1992, the contract Scope of Work for Phase 3 was modified to place greater emphasis on refinement of computational tools initially developed in Phases 1 and 2, and to support evaluation of recently acquired Technology Test Bed (TTB) data. The primary effort during Phase 3 was continued development of software intended to support integration of TTB test data with SSME performance predictions from the steady-state power balance model. In addition, variational analyses of both TTB test data and PBM predictions were performed. Results of these analyses were compared to evaluate the computational integrity of the SSME steady-state power balance model.

A considerable portion of the contract effort was dedicated to development and testing of a formal strategy for reconciling uncertain test data with physically limited computational prediction. This emphasis was motivated by the availability of an extensive and highly organized SSME performance data base from the Engine 3001 test program, and by serious inconsistencies in power balance model predictions.

A review of SSME steady-state power balance model function is provided in Section 2 of this report. Specific weaknesses in the logical structure of the current PBM version are described with emphasis given to the main routing subroutines BAL and DATRED. Selected results from a variational analysis of PBM predictions are compared to TTB variational study results to assess PBM predictive capability.

The motivation for systematic integration of uncertain test data with computational predictions based on limited physical models is provided in Section 3. The theoretical foundation for the reconciliation strategy developed in this effort is presented, and results of a reconciliation analysis of the SSME high pressure fuel side turbopump subsystem are examined. Specific recommendations are presented in Section 4.

2.0 SSME PBM LOGIC ASSESSMENT

The Space Shuttle Main Engine power balance model is a FORTRAN based software package developed by the Rocketdyne Division of Rockwell International. It is used to predict operating characteristics and performance of the SSME under steady-state conditions. Approximately 800 SSME temperatures, pressures, flow rates, shaft speeds, and other hardware performance parameters are calculated by the power balance model.

The current version of PBM has a number of analysis options. The standard power balance analysis option determines fluid and flow properties throughout the entire engine system assuming nominal hardware performance characteristics. In addition, there is a data reduction analysis option which uses actual test data to define hardware operating characteristics such as efficiencies and flow multipliers of a specific SSME. The base balance option is used to further define engine hardware characteristics by matching nine parameters to data reduction output.

Although conceptually a powerful prediction tool, PBM exhibits a number of significant shortcomings. Documentation of the computational, physical, and functional operation of PBM has not been rigorously maintained and is inadequate. Moreover, recent tests have demonstrated that PBM predictions fail to satisfy fundamental energy balance relations within all engine subsystems [1]. As a result, confidence in PBM predictions has been degraded and software utility diminished.

In order to assess the logical integrity of PBM, five sources of information were utilized:

1) detailed flowcharts of the main analysis routing routines BAL and DATRED
2) iteration loop diagrams for routines BAL and DATRED
3) detailed translation of routines BAL and DATRED
4) direct source code inspection
5) analysis of variations (ANOVA) comparisons of PBM predictions and TTB data.

An overview of the main routing logic was obtained from sources 1 and 2. The physical and logical consistency of individual lines of code was examined using sources 3 and 4. The integrity of the PBM prediction process was evaluated using comparisons from source 5.

Detailed, computer-generated flowcharts of PBM subroutines BAL and DATRED were obtained from NASA/MSFC/EP52 and examined for logical structure. Copies of these flowcharts were previously presented in the Phase 2 final report. A high level of vertical connectivity (logic feedback) is obvious upon examination of the complex BAL flowchart. Subroutine logic is highly integrated and structured segmentation cannot be achieved without fundamental and costly code modifications beyond the scope of this effort. Subroutine DATRED has a more sequential logic process, however, structured segmentation of the code was not attempted for reasons described below.

Iteration loop logic diagrams for subroutines BAL and DATRED were also presented in the Phase 2 final report. Subroutine BAL contains four multivariate iteration loops for solution of simultaneous nonlinear relations, and twenty-five univariate iteration loops for solving individual nonlinear relations, all of the Newton-Raphson type. Five deep nesting of iteration sequences is found in BAL with high level multivariate iteration loops traversing virtually the entire routine. Intersection of major iteration sequences inhibits structuring of code logic. Iteration nesting and crossover are not as severe in subroutine DATRED, however, both BAL and DATRED use a segmented solution strategy on restricted subsets of the fluid and flow governing equations. Values of the subset solution variables are iteratively matched with both nested and sequential subset solutions. This type of segmented solution approach with matching is generally less efficient than robust global strategies for solution of nonlinear equations [see, e.g. 2].

In order to facilitate interpretation of PBM logic, a software translator package was constructed in the C programming language. The translator substitutes variable definitions found in PBM documentation in place of the variable names in PBM. The result is a readable document describing PBM function in detail. Translations of subroutines BAL and DATRED were included in the Phase 2 final report. These translations were used to study software logic line by line.

A detailed examination of BAL and DATRED logic indicates that many computations are empirical and/or heuristic. This conclusion is based on comparison of the actual number of SSME flow network controllers with the number of independent variables used by PBM to predict a variety of engine operating conditions. Since the SSME is a feedback dominated flow network, each control setting can be expected to affect operating characteristics throughout the engine. However, many PBM computations are based on reduced dependencies. This is especially evident in the data reduction routine DATRED where certain densities, temperatures, flow rates, pressures, and hardware characteristics are specified by relations depending on thrust level (or commanded chamber pressure) alone.

In order to assess the fundamental dependencies of PBM computations, a variational study of power balance predictions was performed. PBM analyses of engine number 3001 were performed by John Butas of the MSFC Propulsion Laboratory. Analysis independent parameters were set at values corresponding to control parameter settings for each of the 16 test profiles employed in the TTB Engine 3001 test program. Control parameter definitions and settings for each of the 16 TTB program tests are shown in Appendix B, Table 1.

The Engine 3001 test series was based on a Taguchi type design of experiments [3]. The matrix of control settings displayed in Table 1 was selected based on a fractional factorial test plan to facilitate data utilization. A variational analysis of TTB recorded engine operating conditions at test matrix control settings was also performed. Results of the TTB data variational study were then compared to the PBM analysis of variations. Selected parameter comparisons are displayed in Appendix A, Figures 1 through 4. The computed contribution of each individual control parameter to the variation of the performance variable listed in the title is displayed. The abscissa designations OSP, HSP, ORP, and HRP correspond to LOX NPSP, FUEL NPSP, LOX REPRS, and FUEL REPRS respectively in the test matrix. The category COMB that appears in each figure represents contributions from control parameter combinations that cannot be individually allocated because the test program was not designed as a full factorial set of experiments.

It is obvious from Figure 1 that, within the TTB test range, low pressure fuel pump (LPFP) speed variation was only weakly affected by F7 orifice size. PBM predicted F7 contributions to LPFP speed variation were significantly greater than test results indicated. Similarly, as shown in Figure 2, TTB data indicated a significant F7 orifice size contribution (17%) to high pressure fuel pump (HPFP) discharge temperature variation that was largely absent from PBM predictions. Comparisons such as these are indicative of potential PBM weakness in assigning component level contributions to performance and in predicting operational contributions of hardware modification.

Large differences (approximately 31%) between observed and predicted power level (% RPL) and mixture ratio (M/R) contributions to coolant control valve (CCV) flow rate are indicated in Figure 3. Significant disparity between predicted and observed controller effects was not isolated to the parameters displayed in Figures 1 through 3. In general there was good agreement between predicted and observed controller contributions to pressure variation such as is shown in Figure 4 for the high pressure fuel pump (HPFP) outlet pressure. Only isolated cases of significant pressure variational differences were observed. More common were large differences between predicted and observed contributions to temperature, flow rate, and hardware performance characteristics.

The variational analysis comparisons described above suggest that PBM does not adequately model the variation of important SSME performance parameters. This is not particularly surprising since, in many places throughout the code, physical dependencies have been replaced by "hardcoded" empirical relations that lack adequate documentation to assess application and range validity. These comparisons also suggest that the power balance model would be inadequate as a design or anomaly resolution tool. Integrity of PBM predictions can be expected only in nominal engine operating ranges over which code empirical relations were established.

Because the power balance model is a highly connected software package with significant iteration looping, it is difficult to access the overall impact of an individual code modification without significant computational testing. Simple code corrections to achieve improvements in isolated parameter prediction can have a far reaching and detrimental affect. Therefore, code maintenance and modification time will be substantially greater than for a highly structured, modular, and well documented performance model.

One of the major functions of the power balance model is to integrate test data with theoretical predictions. The weaknesses of the existing data integration procedure will be discussed in the following section, and a new integration strategy will be introduced.

3.0 RECONCILIATION MODEL

One of the features of the steady-state power balance model is its ability to integrate test data into the performance prediction process. This is accomplished within the data reduction analysis option. In the data reduction process, test information is incorporated literally into predictions, and hardware operating parameters are adjusted to values consistent with this presumed pristine data.

Unfortunately, experimental data associated with a complex flow system such as the SSME is fraught with uncertainty. Maintaining operation and calibration of sensing and signal conditioning instrumentation is difficult in the severe SSME operating environment. In addition, point measurements in such a complex flow environment often include the effects of highly localized and/or secondary flow phenomena that are not characteristic of system average conditions. Literal incorporation of inaccurate test data can lead to nonphysical predictions of engine operation and erroneous assumptions concerning hardware performance. Since all test data has associated uncertainty, the pristine data assumption is inappropriate in a test information integration strategy.

Performance prediction models based on fundamental flow physics are also limited by theoretical approximations required to achieve tractable solution. For example, PBM computations assume steady-state operation throughout the engine, and provide estimates of average flow conditions using a cross-stream uniform, one-dimensional flow approximation. In addition, thermodynamic property data for hydrogen and oxygen in SSME operating ranges has accuracy limitations. These type assumptions and limitations necessarily restrict the accuracy of theoretical model predictions and present an additional source of uncertainty for data integration strategies.

The above observations suggest that any systematic procedure for integrating experimental data and theoretical predictions should recognize both data uncertainty and model limitations. The objective of the reconciliation development effort undertaken as part of this study was to construct a logical strategy for integrating uncertain test data with limited theoretical predictions in order to determine most plausible SSME operating conditions.

A heuristic yet logical procedure for achieving systematic data integration was presented in the Phase 1 final report of this study. A refinement of this reconciliation procedure has been developed in Phase 3. The basis of the new method rests on the principle that the mean of experimental observations reflects most probable, but not absolute, engine operating conditions. If measured engine operating properties are assumed to be independent, normally distributed, random variables, then the most probable engine state will maximize the property joint probability density function (pdf) subject to constraints imposed by physical laws. A mathematical expression for this state pdf is given below:

$$F(X_1 \ldots X_k) = \frac{1}{\sigma_1 \ldots \sigma_k (2\pi)^{k/2}} e^{-\left[\frac{d_1^2}{2\sigma_1^2} + \ldots + \frac{d_k^2}{2\sigma_k^2}\right]} \tag{1}$$

where

X_i - adjusted value of property i
σ_i - standard deviation of property i
μ_i - mean of property i
d_i - deviation of adjusted property i from its mean (measured value) ($d_i = X_i - \mu_i$)
F - joint probability density function of state
k - number of properties
$(X_1 \ldots X_k)$ - state of system

Properties in the relation above include measured flow rates, temperatures, and pressures throughout the engine system.

The state pdf is a maximum when the expression in brackets in the exponent of e is minimized. In the absence of physical constraints this minimum would occur when all the d_i are zero, or when the value of each property i is at its mean. Therefore, experimental property measurements are assumed to correspond to the property means in equation 1, and the d_i are adjustments from measurement values required to adequately satisfy physical constraints, including mass and energy conservation requirements as well as second law limits.

A robust data reconciliation strategy must also incorporate measurement system uncertainty limits in addition to physical constraints. The problem of determining most plausible engine operating conditions can thus be reduced to a mathematical programming problem of the form:

$$\text{maximize} \quad Z = F(d_1 \ldots d_k) \qquad \begin{array}{l}\text{by selection of} \quad (d_1 \ldots d_k) \\ \text{assuming constant} \ (\sigma_1 \ldots \sigma_k)\end{array}$$

subject to

physical constraints for each engine subsystem j

$$| \text{ mass flow imbalance}_j \ | \quad < \quad L_{flow-j}$$

$$| \text{ energy imbalance}_j \ | \quad < \quad L_{energy-j}$$

$$| \text{ entropy production}_j \ | \quad > \quad 0$$

uncertainty limits for measurements at each node i

$$| \text{ mass flow adjustment}_i \ | \quad < \quad U_{m-i}$$

$$| \text{ pressure adjustment}_i \ | \quad < \quad U_{p-i}$$

$$| \text{ temperature adjustment}_i \ | \quad < \quad U_{T-i} \tag{2}$$

where

$$\text{mass flow imbalance} = \text{ImbM} = \sum_{\text{inlets}} m \quad + \sum_{\text{outlets}} m$$

$$\text{energy imbalance} = \text{ImbE} = \sum_{\text{inlets}} m \, [h + m^2/(2\rho^2A^2)] - \sum_{\text{outlets}} m \, [h + m^2/(2\rho^2A^2)] + Q - W$$

$$\text{entropy production} = \text{ImbS} = \sum_{\text{inlets}} m \, [\ s\ ] - \sum_{\text{outlets}} m \, [\ s\ ] + Q/T_0$$

d_{m-i} - mass flow adjustment at node i

d_{p-i} - pressure adjustment at node i

d_{T-i} - temperature adjustment at node i

L_{flow-j} - upper limit of flow imbalance for flow circuit j

$L_{energy-j}$ - upper limit of energy imbalance for control volume j

U_{m-i} - flow uncertainty limit at node i

U_{p-i} - pressure uncertainty limit at node i

U_{T-i} - temperature uncertainty limit at node i

m - mass flow rate across inlet or outlet surface

h - specific enthalpy at inlet or outlet surface

s - specific entropy at inlet or outlet surface

ρ - mass density at inlet or outlet surface

A - cross-sectional area of inlet or outlet surface

Q - rate of energy transfer as heat into control volume

W - rate of energy transfer as work from control volume

T_0 - temperature at which energy as heat enters volume

The mathematical programming problem stated in formulation 2 above is highly nonlinear. Without loss of generality, the objective function Z = F can be replaced by the exponent of e in equation 1. If, in addition, the imbalance relations are approximated as first order truncated Taylor series expansions in the nodal adjustment values d, the mathematical programming problem reduces to the following:

$$\text{minimize} \quad Z = \sum_{i=1}^{k} \frac{d_i^2}{2\sigma_i^2} \qquad k = \text{number of measurements}$$

subject to

linearized forms of the physical constraints for each engine subsystem j

$$| \, LImbM(\mathbf{d})_j \, | < L_{flow-j}$$

$$| \, LImbE(\mathbf{d})_j \, | < L_{energy-j}$$

$$| \, LImbS(\mathbf{d})_j \, | > 0$$

measurement uncertainty limits for each node i
(n = number of nodes = number of measurements/3 = k/3)

$$| \, d_{m-i} \, | < U_{m-i}$$

$$| \, d_{p-i} \, | < U_{p-i}$$

$$| \, d_{T-i} \, | < U_{T-i} \qquad (3)$$

where

$$\mathbf{d} = \begin{matrix} d_1 \\ \cdot \\ \cdot \\ d_k \end{matrix} = \begin{matrix} d_{m-1} \\ \cdot \\ \cdot \\ d_{m-n} \\ d_{p-1} \\ \cdot \\ \cdot \\ d_{p-n} \\ d_{T-1} \\ \cdot \\ \cdot \\ d_{T-n} \end{matrix}$$

The objective function Z in formulation 3 above is quadratic in the measurement adjustments **d**, and the constraints have been linearized in terms of the adjustment variables **d**. This is the form of a classical quadratic programming problem for which a variety of robust solution strategies exist. The solution of this problem minimizes the property adjustments required to satisfy physical constraints within measurement system uncertainty bounds, and in a logical sense provides most plausible engine operating conditions subject to restrictions inherent in the linearization of the physical constraints.

The reconciler developed as part of this effort constructs the quadratic programming problem defined in formulation 3 above and implements the complementary pivot method algorithm [4] to obtain the QP problem solution. A hierarchy diagram describing the organization of routines in the reconciler is presented in Appendix A, Figure 5. Documentation describing the function of reconciler routines is given in Appendix C1, and a source code listing is presented in Appendix C2.

In order to perform a reconciliation analysis, four types of input data are required. Thermodynamic property data in operating ranges of interest are necessary. Specific enthalpy, specific entropy, and density as functions of pressure and temperature are required. For SSME analyses, hydrogen, oxygen, and water property information was provided and integrated into the reconciler logic by John Butas of NASA/MSFC/EP52. In addition, experimental data (or computational simulation results) are required to provide a baseline for adjustment calculations. The TTB Engine 3001 test program has provided extensive high quality experimental data for reconciliation analyses. PBM predictions have provided a simulation baseline for initial reconciler testing. The third type of input required for reconciliation analysis is uncertainty estimates quantifying model limitations (physical constraint bounds

L in formulation 3) as well as test data confidence bands (uncertainty bounds U in formulation 3). Finally, system definition information must be constructed to specify engine configuration and to properly associate nodal property data. A detailed description of input data requirements is provided in Appendix C1 documentation and a listing of the reconciler input data file format is provided in Appendix B, Table 2.

Reconciler performance has been verified on a number of test problems. Recently, reconciler logic was tested using results of a PBM simulation of the HPFTP system at 109% RPL to provide baseline measurements. A schematic of the HPFTP system with analysis nodes identified is displayed in Appendix A, Figure 6. The analysis configuration was composed of 14 nodes, 5 mass flow circuits, and 4 energy volumes. Mass circuit and energy volume definition nodes for this analysis are specified in Appendix B, Table 3. The energy volumes include the fuel preburner, high pressure fuel turbopump, fuel side turn around duct, and fuel side hot gas manifold. Coarse measurement system uncertainty estimates were utilized in the HPFTP test case analysis because more precise uncertainty information was unavailable. These estimates are provided in Appendix B, Table 4.

Mass, energy, entropy, and availability imbalances both before and after reconciliation are displayed in Appendix B, Tables 5 and 6 respectively. A significant reduction in subsystem energy imbalances after reconciliation is the most striking result observed in comparing Tables 5 and 6. A 99% energy imbalance reduction in the fuel preburner and turbopump subsystems was obtained during the reconciliation process while mass balance and entropy production requirements were maintained. System properties both before and after reconciliation are presented in Appendix B, Table 7. The adjustments required to achieve reconciliation (solution to the quadratic programming problem outlined in formulation 3) are also presented in Table 7. Significant reductions in PBM predicted hot gas temperatures throughout the system are suggested. These temperature reductions remain within specified measurement uncertainty bounds, yet provide substantial improvement in the initially large energy imbalances.

Heuristic data integration procedures do not provide the level of confidence in prediction that is required in a long term engine development program. Efficient and reliable use of experimental observation to improve performance prediction and safety requires a systematic test data integration strategy. The reconciliation strategy outlined above is a logical procedure for improving rocket engine performance prediction. The mathematical foundation is well defined and computations are physically sound within approximation limits. In addition, the base procedure is completely general, with material and system configuration provided by modular data file inputs. Initial test results have been quite successful and strongly support continued development and use of this mathematical programming approach for large system test data reconciliation. This technique can also be utilized to evaluate test data integrity and isolate measurement system problems.

4.0 RECOMMENDATIONS

A list of recommendations based on results of this research effort is presented below.

1. Local modifications to the power balance model should be thoroughly investigated before implementation due to the high level of logic connectivity. If PBM is to be used extensively as a prediction tool, a catalog of parameter influence coefficients should be developed to efficiently assess the system wide impact of specific code changes.

2. Without extensive documentation describing imbedded empiricisms within PBM logic, the power balance model should be considered a high order gains model containing the experience base archive. PBM should not be considered a cornerstone theoretical prediction tool without major modification.

3. Development of mathematical programming approaches to test data reconciliation should continue in order to provide a consistent and logical basis for improving performance prediction, a platform for logically resolving data inconsistencies, and a means of assessing data and measurement system integrity.

4. A fundamentally sound theoretical model of engine performance should be developed.

5. Uncertainty analysis should be incorporated in any rocket engine performance evaluation and prediction program.

6. An integrated rocket engine performance prediction platform should be developed that modularizes fundamental theoretical computations and provides a standardized interface for

efficient parametric integration of test data.

7. Frictional resistance relations should be added to the quadratic reconciler in order to provide more consistent pressure loss relations.

5.0 REFERENCES

1. Santi, L. M., "Validation of the Space Shuttle Main Engine Steady State Performance Model," NASA Contractors Report CR-18404-XLI, Oct., 1990.

2. Rheinboldt, Werner C., Methods for Solving Systems of Nonlinear Equations. SIAM, CBMS-NSF Regional Conference Series in Applied Mathematics, Philadelphia, 1974.

3. Taguchi, Gen'ichi, System of Experimental Design: Engineering Methods to Optimize Quality and Minimize Cost. UNIPUB/Kraus International Publications, 1989.

4. Ravindran, A., "A Computer Routine for Quadratic and Linear Programming Problems," ACM-Comm, Sept., 1972, pp.818-820.

APPENDIX A

FIGURES

Figure 1. LPFP Speed

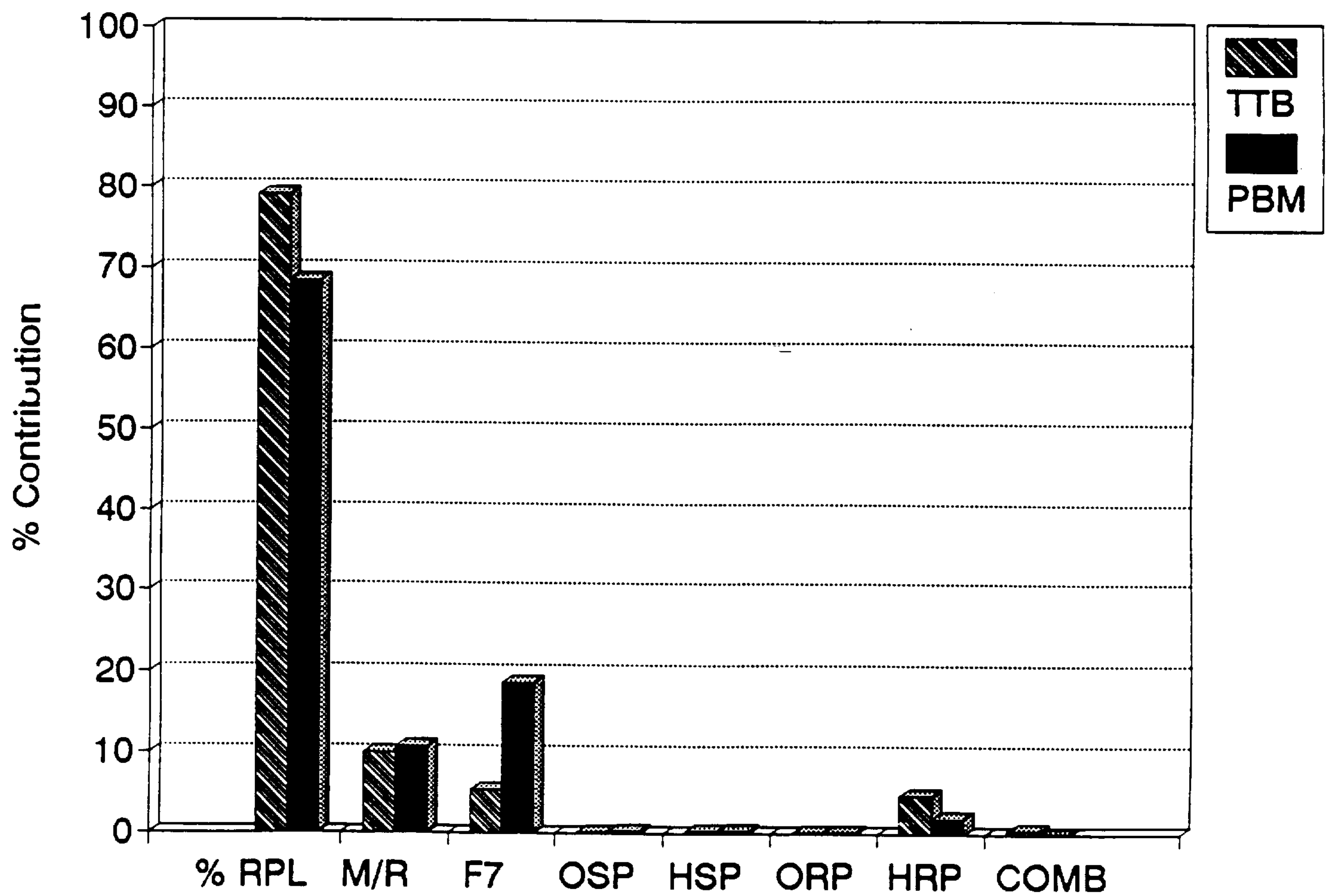


Figure 2. HPFP Disch Temp

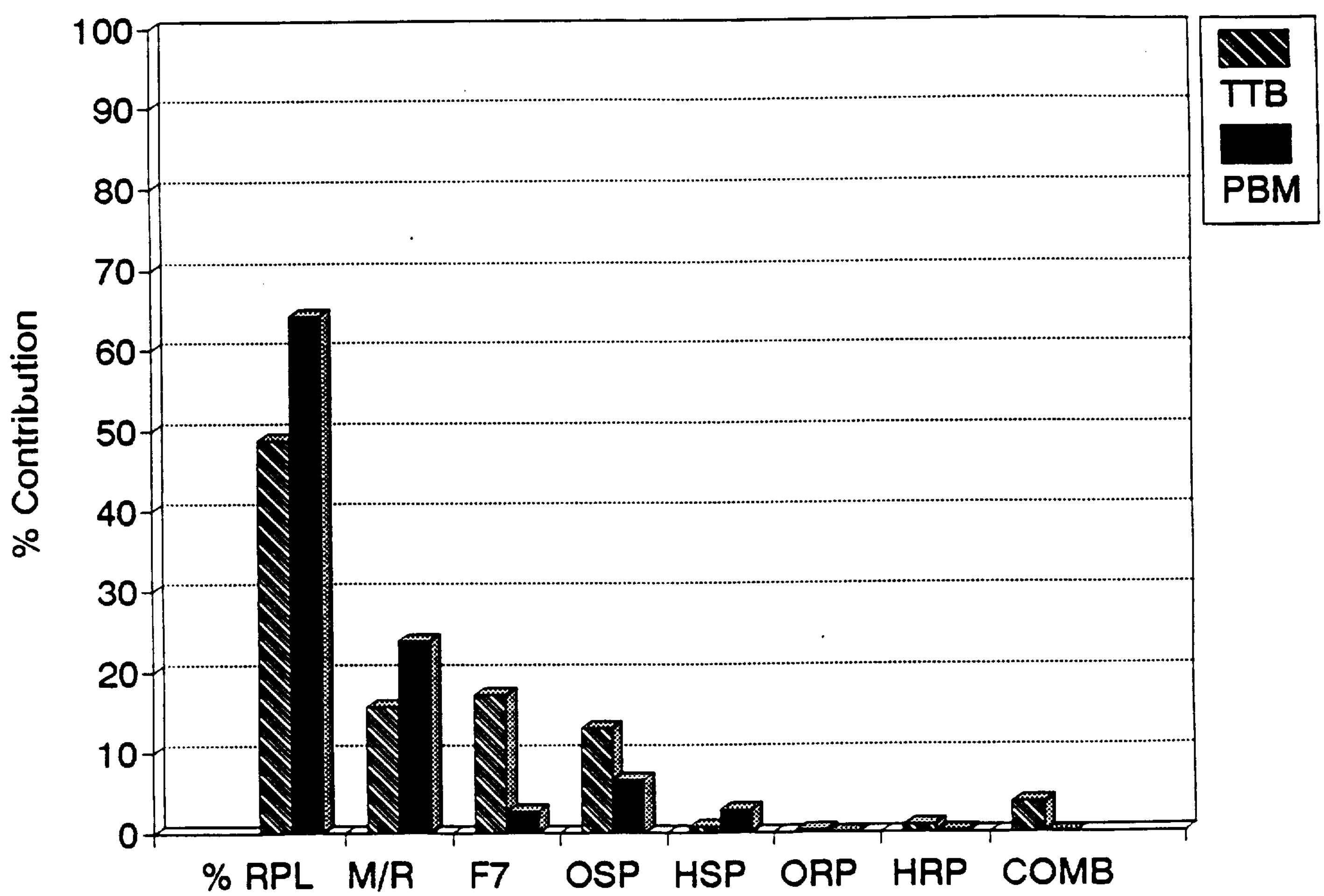


Figure 3. CCV Flow Rate

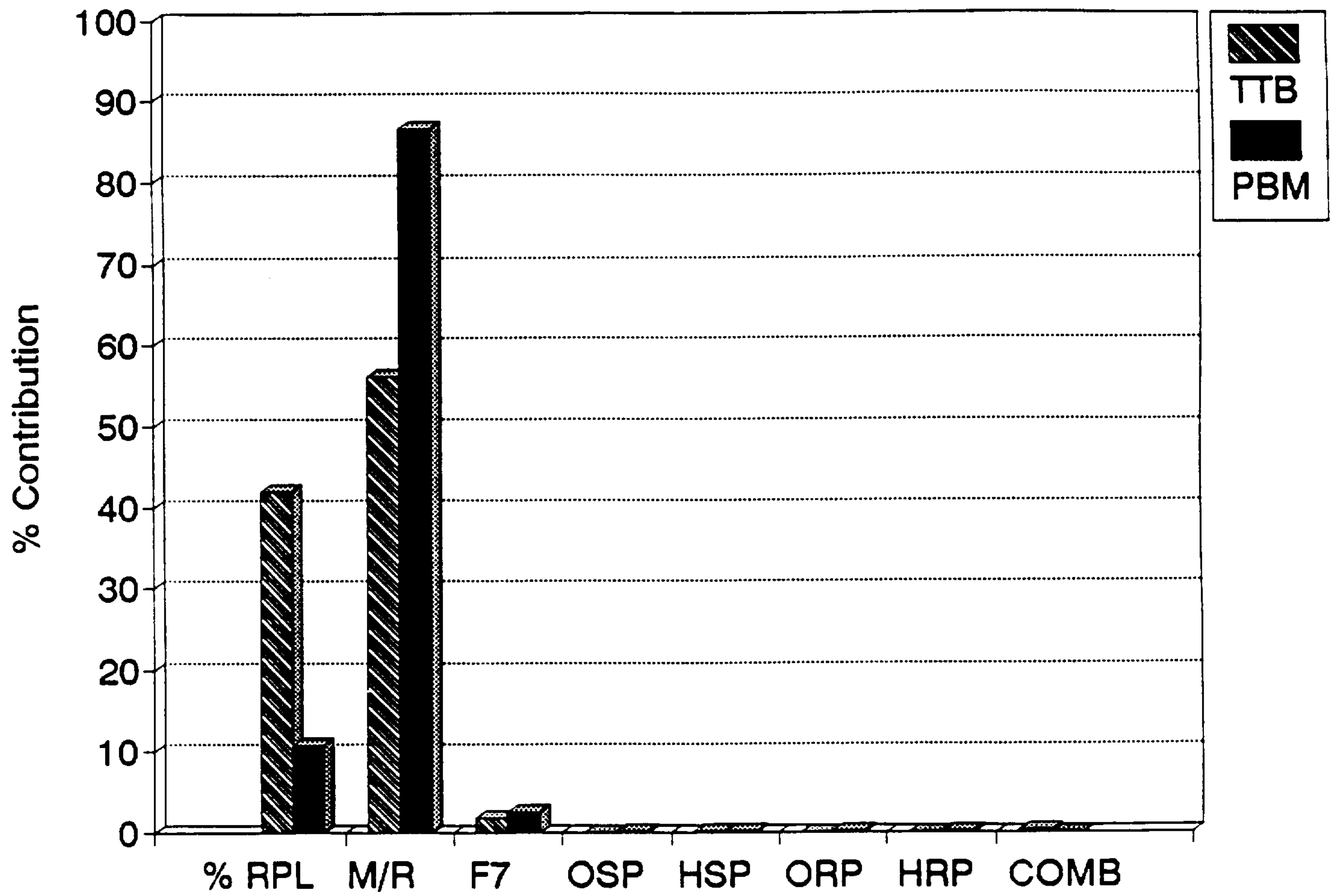


Figure 4. HPFP Disch Pressure

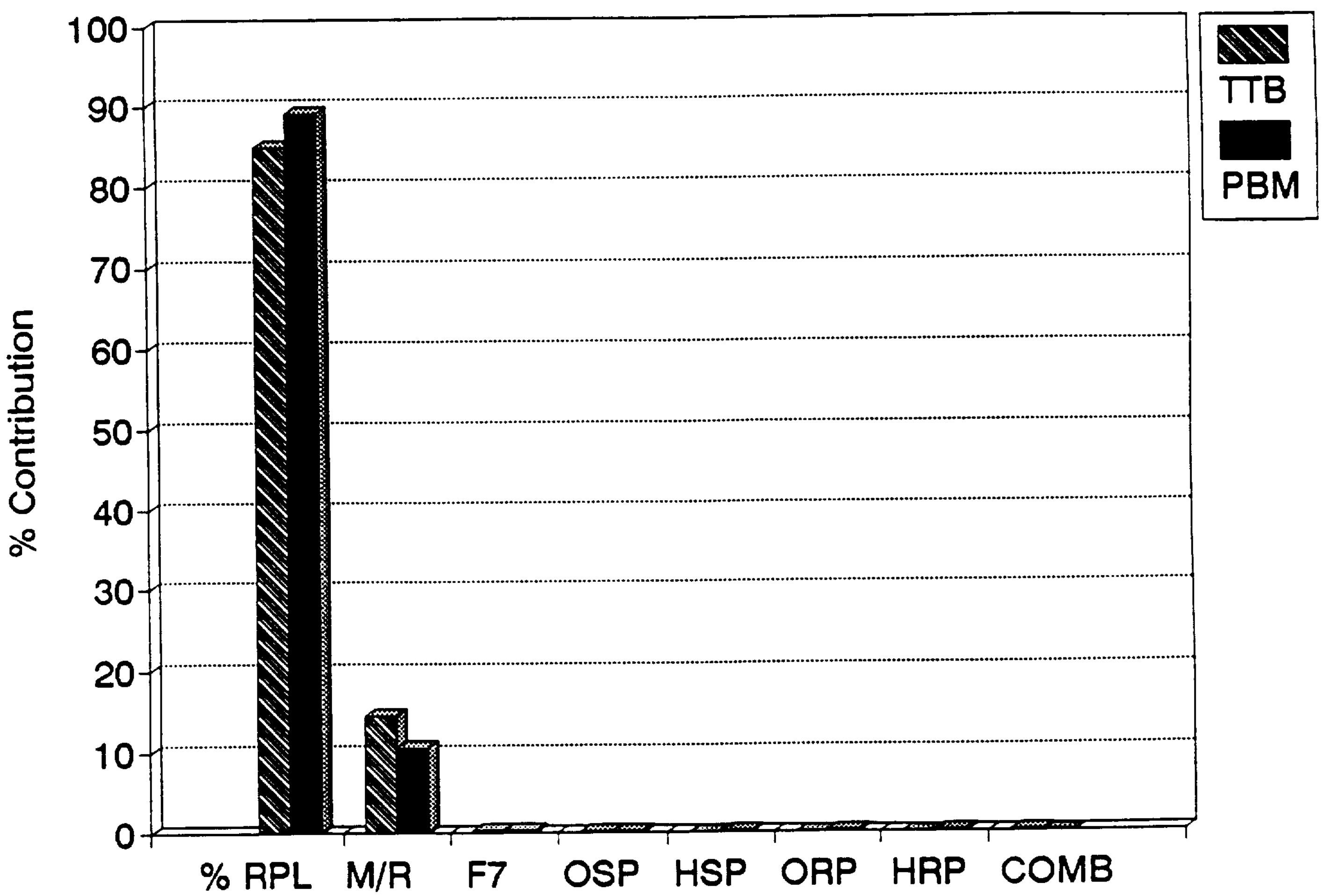
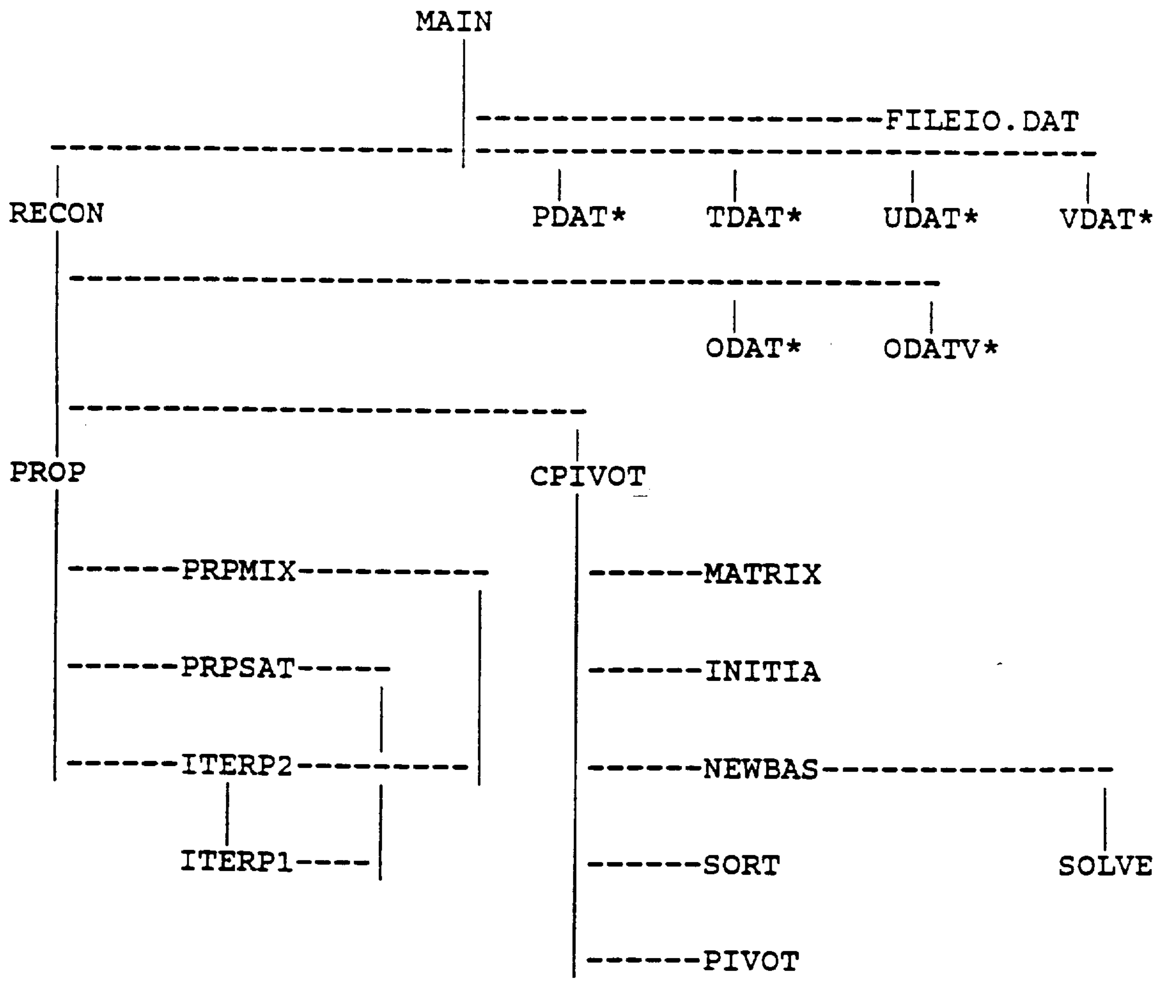
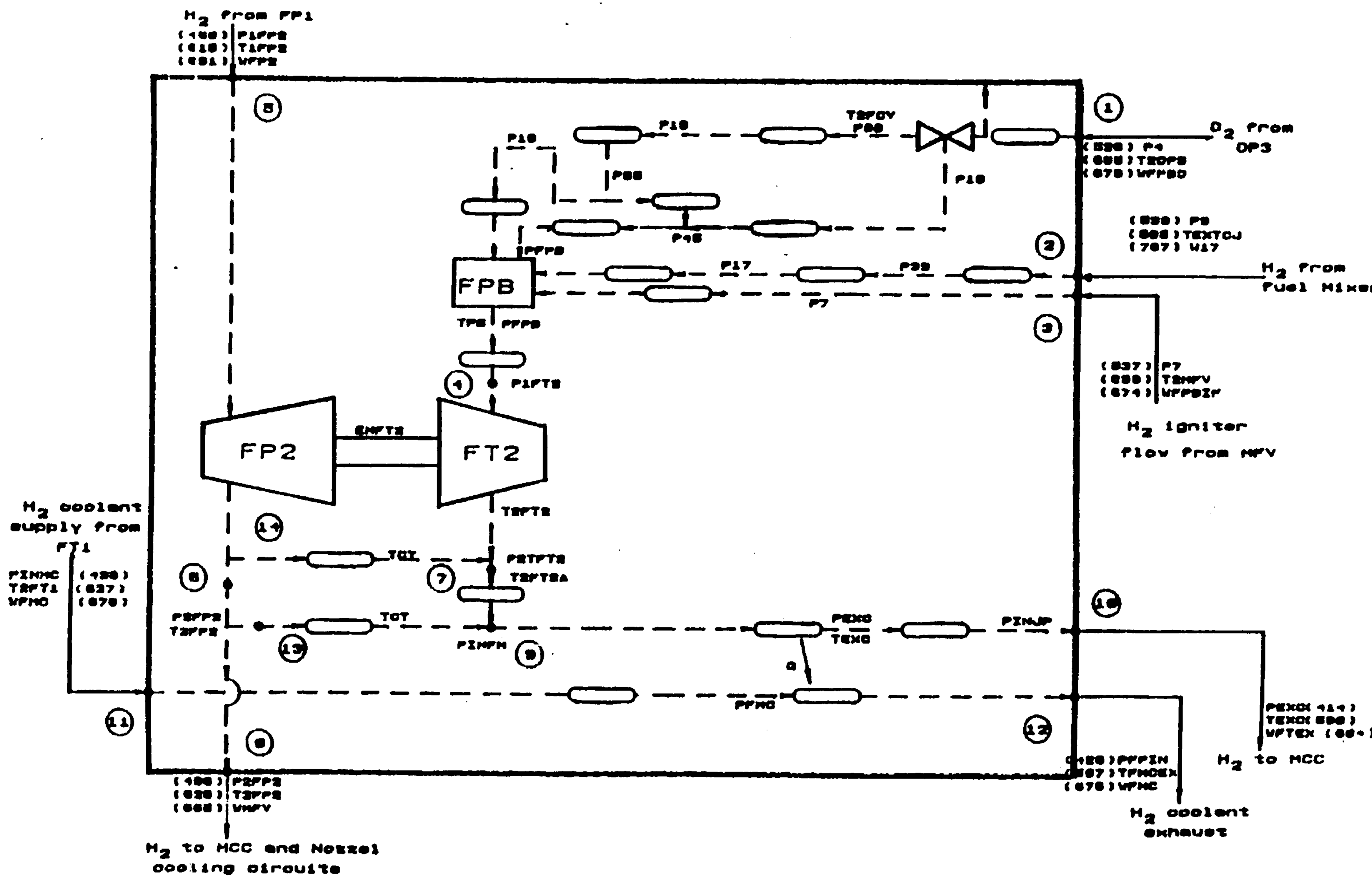


Figure 5. Quadratic Reconciler Hierarchy Diagram



* - indicates data file designated in FILEIO.DAT

Figure 6. HPFTP system with reconciliation analysis nodes indicated



APPENDIX B

TABLES

Table 1. TTB Engine 3001 test program matrix of control variable settings

TEST PT	POWER LEVEL	MR	F7	LO2 NPSP	LH2 NPSP	GO2 REPRS	GH2 REPRS	TEST #	TIME SLICE
1	100.00	5.85	1.70	20.00	6.00	1.10	0.20	TTB-25	109-119
2	104.00	5.85	1.70	20.00	24.50	2.40	1.20	TTB-25	134-139
3	100.00	5.85	1.70	120.00	24.50	2.40	0.20	TTB-25	30-35
4	104.00	5.85	1.70	120.00	6.00	1.10	1.20	TTB-25	49-59
5	100.00	5.85	1.90	20.00	24.50	1.10	1.20	TTB-28	114-119
6	104.00	5.85	1.90	20.00	6.00	2.40	0.20	TTB-28	84-89
7	100.00	5.85	1.90	120.00	6.00	2.40	1.20	TTB-28	30-35
8	104.00	5.85	1.90	120.00	24.50	1.10	0.20	TTB-27	36-41
9	100.00	6.15	1.70	20.00	24.50	2.40	1.20	TTB-25	154-159
10	104.00	6.15	1.70	20.00	6.00	1.10	0.20	TTB-25	194-199
11	100.00	6.15	1.70	120.00	6.00	1.10	1.20	TTB-23	69-74
12	104.00	6.15	1.70	120.00	24.50	2.40	0.20	TTB-23	40-45
13	100.00	6.15	1.90	20.00	6.00	2.40	0.20	TTB-26	118-122
14	104.00	6.15	1.90	20.00	24.50	1.10	1.20	TTB-26	144-154
15	100.00	6.15	1.90	120.00	24.50	1.10	0.20	TTB-26	30-35
16	104.00	6.15	1.90	120.00	6.00	2.40	1.20	TTB-26	59-64

DEFINITIONS			Figure Designation
Power Level	-	percent of SSME rated power level	% RPL
MR	-	mixture ratio (mass O2)/(mass H2)	M/R
F7	-	F7 orifice area (square inches)	F7
LOX NPSP	-	liquid oxygen net positive suction pressure (psi)	OSP
Fuel NPSP	-	liquid hydrogen net positive suction pressure (psi)	HSP
LOX REPRS	-	oxygen repressurization flow rate (lb/s)	ORP
Fuel REPRS	-	hydrogen repressurization flow rate (lb/s)	HRP

Table 2. Reconciler input data file formats

```
FILEIO.DAT  -  designates I/O data filenames

     'input property data filename'
     'input test data filename'
     'input uncertainty estimates filename'
     'input volume definition data filename'
     'output (standard format) filename'
     'output (test data input format) filename'


TDAT = 'input test data filename'

     NDESC   NTTB
     DESC(1).........DESC(NDESC)
     TTB(1)..........TTB(NTTB)


UDAT = 'input uncertainty estimates filename'

     IPRPD   ITTBD   IROWA   JCOLA   ITORDX   ITORDY   DPF   DTF
     UP(1)...........UP(NTNOD)
     UT(1)...........UT(NTNOD)
     UW(1)...........UW(NTNOD)
     UWMFC(I)........UWMFC(NTMFC)
     UEVOL(1)........UEVOL(NTVOL)
     USVOL(1)........USVOL(NTVOL)
```

(continued next page)

Table 2. Reconciler input data file formats

VDAT = 'input volume definition data filename'

```
IENV   IPCTTH   MAXSTG   NHG   NTMFC   NTNOD   NTVOL
IA(1)...............IA(NTNOD)
IP(1)...............IP(NTNOD)
IT(1)...............IT(NTNOD)
IW(1)...............IW(NTNOD)
MAT(1)..............MAT(NTNOD)
MIO(1)..............MIO(NTMFC)
NIO(1)..............NIO(NTVOL)
MODIR(1,1)..........MODIR(1,MIO(1))
.     IMFCN(1,1)..................IMFCN(1,NTMFC))
.                                                  .
.                                                  .
MODIR(NTMFC,1).....MODIR(NTMFC,MIO(NTMFC))         .
      IMFCN(NTMFC,1)..............IMFCN(NTMFC,MIO(NTMFC))
IODIR(1,1)..........IODIR(1,NIO(1))
.     IVOLN(1,1)..................IVOLN(1,NTVOL))
.                                                  .
.                                                  .
IODIR(NTVOL,1).....IODIR(NTVOL,NIO(NTVOL))         .
      IVOLN(NTVOL,1)..............IVOLN(NTVOL,NIO(NTVOL))
NODHG(1)............NODHG(NHG)
NH2HG(1)............NH2HG(NHG)
NO2HG(1)............NO2HG(NHG)
ICEFF(1)............ICEFF(NHG)
IH2HG(1,1)..........IH2HG(1,NH2HG(1))
.     IO2HG(1,1)................IO2HG(1,NO2HG(1))
.                                                  .
.                                                  .
IH2HG(NHG,1)........IH2HG(NHG,NH2HG(NHG))          .
      IO2HG(NHG,1)..............IO2HG(NHG,NO2HG(NHG))
```

Table 3. HPFTP analysis number 1 circuit definitions

Energy Volume Definitions

Volume #	Boundary Nodes
1	1 2 3 4
2	4 5 6 7
3	6 7 8 9
4	9 10 11 12

Mass Circuit Definitions

Circuit #	Boundary Nodes
1	1 2 3 4
2	5 6 14
3	4 7 14
4	6 8 13
5	7 9 13

Table 4. Uncertainty estimates for HPFTP analysis number 1

Nodal Property Uncertainty

Node #	Pressure (psi)	Temperature (degF)	Flow Rate (lb/sec)
1	0.772	0.0208	0.00796
2	0.605	0.1	0.00831
3	0.659	0.00972	0.0001
4	200.0	100.0	1.6
5	0.0272	0.00424	0.0162
6	200.0	20.0	1.7
7	200.0	100.0	1.6
8	200.0	20.0	1.6
9	200.0	100.0	1.7
10	200.0	100.0	1.7
11	0.359	0.0436	0.00175
12	200.0	200.0	0.00175
13	200.0	20.0	0.14
14	200.0	20.0	0.3

Mass Imbalance Limits

Circuit #	Flow Imbalance Limit (lb/sec)
1	0.001
2	0.001
3	0.001
4	0.001
5	0.001

Volume Imbalance Limits

Volume #	Energy Imbalance Limit (Btu/sec)	Entropy Imbalance Limit (Btu/degR-sec)
1	71.0	100.0
2	43.0	100.0
3	2.6	100.0
4	18.0	100.0

Table 5. High pressure fuel turbopump system imbalances at 109% RPL prior to reconciliation

HP_FUEL_SIDE_ANALYSIS_1

THRUST = 109.0% RPL

SUBSYSTEM	DW(LB/S)	DE(BTU/S)	DE(HP)	DS(BTU/R-S)	DAV(BTU/S)
FPB_SUBSYSTEM	.014	-7111.634	-10061.800	-664.165	344895.60
HPFTP_SUBSYSTEM	.000	-4277.324	-6051.713	-174.517	88216.79
TAD_SUBSYSTEM	.100	-2563.647	-3627.140	-12.175	3889.41
HGM_SUBSYSTEM	.000	183.112	259.074	-5.406	3048.38

Table 6. High pressure fuel turbopump system imbalances at 109% RPL after reconciliation

HP_FUEL_SIDE_ANALYSIS_1

THRUST = 109.0% RPL

SUBSYSTEM	DW(LB/S)	DE(BTU/S)	DE(HP)	DS(BTU/R-S)	DAV(BTU/S)
FPB_SUBSYSTEM	.002	-59.775	-84.571	-660.372	349937.30
HPFTP_SUBSYSTEM	-.003	-17.536	-24.811	-165.596	87748.05
TAD_SUBSYSTEM	.004	-58.564	-82.858	-6.975	3638.13
HGM_SUBSYSTEM	.024	-227.164	-321.400	-.467	20.30

Table 7. HPFTP System Reconciliation at 109% RPL

HPFTP ANALYSIS NO 1

ORIGINAL NODE DATA

NODE	PRESSURE	TEMPERATURE	FLOWRATE
1	7723.00	208.30	79.63
2	6049.00	278.40	83.07
3	6585.00	97.16	1.01
4	5498.00	1929.00	163.70
5	271.70	42.40	162.30
6	6739.00	96.71	159.20
7	3718.00	1737.00	166.80
8	6739.00	96.71	157.80
9	3691.00	1720.00	168.10
10	3585.00	1715.00	168.10
11	3587.00	436.30	17.51
12	3574.00	460.90	17.51
13	6739.00	96.71	1.38
14	6739.00	96.71	3.05

RECONCILED NODE DATA

NODE	PRESSURE	TEMPERATURE	FLOWRATE
1	7723.00	208.30	79.63
2	6049.00	278.40	83.07
3	6585.00	97.16	1.01
4	5511.55	1907.02	163.71
5	271.70	42.40	162.30
6	6719.12	95.07	159.25
7	3704.84	1704.95	166.76
8	6723.15	93.87	157.87
9	3657.60	1682.59	168.14
10	3569.42	1687.03	168.12
11	3587.00	436.30	17.51
12	3543.89	419.48	17.51
13	6722.71	96.53	1.38
14	6722.71	96.53	3.05

(continued next page)

Table 7. HPFTP System Reconciliation at 109% RPL

HPFTP ANALYSIS NO 1

BALANCING ADJUSTMENTS

NODE	PRESSURE	TEMPERATURE	FLOWRATE
1	.00	.00	.00
2	.00	.00	.00
3	.00	.00	.00
4	13.55	-21.98	.01
5	.00	.00	.00
6	-19.88	-1.64	.05
7	-13.16	-32.05	-.04
8	-15.85	-2.84	.07
9	-33.40	-37.41	.04
10	-15.58	-27.97	.02
11	.00	.00	.00
12	-30.11	-41.42	.00
13	-16.29	-.18	.00
14	-16.29	-.18	.00

PERCENT ADJUSTMENT

NODE	PRESSURE	TEMPERATURE	FLOWRATE
1	.00	.00	.00
2	.00	.00	.00
3	.00	.00	.00
4	.25	-1.14	.01
5	.00	.00	.00
6	-.30	-1.69	.03
7	-.35	-1.84	-.02
8	-.24	-2.94	.04
9	-.90	-2.18	.02
10	-.43	-1.63	.01
11	.00	.00	.00
12	-.84	-8.99	.00
13	-.24	-.18	-.01
14	-.24	-.18	.07

APPENDIX C1

RECONCILER DOCUMENTATION

Routine: MAIN (main program for reconciliation model)

Routine Function:

Main calling routine for reconciliation model. Opens and reads input data files associated with thermodynamic property data, control volume definition, test data, and uncertainty estimates. Initializaes parameters for reconciliation model.

Common Blocks:

Blank - quadratic programming algorithm matrices and parameters
TDAT - test data
UDAT - uncertainty estimates
VDAT - control volume definition information
H2PRP - hydrogen table pressures and temperatures, and associated enthalpies, entropies, and densities
O2PRP - oxygen table pressures and temperatures, and associated enthalpies, entropies, and densities
H2OPRP- water table pressures and temperatures, and associated enthalpies, entropies, and densities
TABLE - number of distinct pressures and temperatures in each hydrogen, oxygen, and water table
STD - reference state enthalpies entropies, and absolute entropies for hydrogen, oxygen, and water

Input Variable Definitions:

Input File 'FILEIO.DAT' (identifies input files)
PDAT - alpha variable that identifies name of property data input file
TDAT - alpha variable that identifies name of test data input file
UDAT - alpha variable that identifies name of uncertainty data input file
VDAT - alpha variable that identifies name of volume defintion input file
ODAT - alpha variable that identifies name of main reconciler output file
ODATV - alpha variable that identifies name of reconciler output file that is in same format as variable TDAT input file

Input File PDAT (property data input file) (file name variable identified in FILEIO.DAT)
PTITLE - property data table subsection title
NH2P(I) - number of pressures in H2 property data table I
NH2T(I) - number of temperatures in H2 property data table I

H2P1(I) - pressure I in H2 data table 1
H2T1(I) - temperature I in H2 data table 1
H2H1(I,J) - H2 enthalpy associated with pressure I and temperature J in H2 data table 1
H2S1(I,J) - H2 entropy associated with pressure I and temperature J in H2 data table 1
H2D1(I,J) - H2 density associated with pressure I and temperature J in H2 data table 1
H2P2(I) - pressure I in H2 data table 2
H2T2(I) - temperature I in H2 data table 2
H2H2(I,J) - H2 enthalpy associated with pressure I and temperature J in H2 data table 2
H2S2(I,J) - H2 entropy associated with pressure I and temperature J in H2 data table 2
H2D2(I,J) - H2 density associated with pressure I and temperature J in H2 data table 2
H2P3(I) - pressure I in H2 data table 3
H2T3(I) - temperature I in H2 data table 3
H2H3(I,J) - H2 enthalpy associated with pressure I and temperature J in H2 data table 3
H2S3(I,J) - H2 entropy associated with pressure I and temperature J in H2 data table 3
H2D3(I,J) - H2 density associated with pressure I and temperature J in H2 data table 3
H2P4(I) - pressure I in H2 data table 4
H2T4(I) - temperature I in H2 data table 4
H2H4(I,J) - H2 enthalpy associated with pressure I and temperature J in H2 data table 4
H2S4(I,J) - H2 entropy associated with pressure I and temperature J in H2 data table 4
H2D4(I,J) - H2 density associated with pressure I and temperature J in H2 data table 4
NO2P(I) - number of pressures in O2 property data table I
NO2T(I) - number of temperatures in O2 property data table I
O2P1(I) - pressure I in O2 data table 1
O2T1(I) - temperature I in O2 data table 1
O2H1(I,J) - O2 enthalpy associated with pressure I and temperature J in O2 data table 1
O2S1(I,J) - O2 entropy associated with pressure I and temperature J in O2 data table 1
O2D1(I,J) - O2 density associated with pressure I and temperature J in O2 data table 1
O2P2(I) - pressure I in O2 data table 2
O2T2(I) - temperature I in O2 data table 2
O2H2(I,J) - O2 enthalpy associated with pressure I and temperature J in O2 data table 2
O2S2(I,J) - O2 entropy associated with pressure I and temperature J in O2 data table 2
O2D2(I,J) - O2 density associated with pressure I and temperature J in O2 data table 2

O2P3(I) - pressure I in O2 data table 3
O2T3(I) - temperature I in O2 data table 3
O2H3(I,J) - O2 enthalpy associated with pressure I and temperature J in O2 data table 3
O2S3(I,J) - O2 entropy associated with pressure I and temperature J in O2 data table 3
O2D3(I,J) - O2 density associated with pressure I and temperature J in O2 data table 3
NH2OP(I) - number of pressures in H2O property data table I
NH2OT(I) - number of temperatures in H2O property data table I
H2OP1(I) - pressure I in H2O data table 1
H2OT1(I) - temperature I in H2O data table 1
H2OH1(I,J) - H2O enthalpy associated with pressure I and temperature J in H2O data table 1
H2OS1(I,J) - H2O entropy associated with pressure I and temperature J in H2O data table 1
H2OD1(I,J) - H2O density associated with pressure I and temperature J in H2O data table 1

Input File TDAT (test data input file)
(file name variable identified in FILEIO.DAT)

NDESC - number of alpha variable test data descriptions
NTTB - number of data entries in test data table TTB(I)
DESC(I) - alpha variable test data description I
TTB(I) - test data entry I

Input File VDAT (volume definition input file)
(file name variable identified in FILEIO.DAT)

IENV - TTB array address of environmental temperature
IPCTTH - TTB array address of % rated power level
MAXSTG - number of stages to be used in SQP iteration sequence
NHG - number of system nodes at which HG flow occurs
NTMFC - number of mass flow circuits in engine system analysis
NTNOD - number of nodes in engine system analysis
NTVOL - number of volumes in engine system analysis
IA(I) - position in TTB array containing the value of the cross-sectional area at node I
IP(I) - position in TTB array containing the value of the pressure at node I
IT(I) - position in TTB array containing the value of - the temperature at node I
IW(I) - position in TTB array containing the value of the mass flow rate at node I
MAT(I) - material identifying number at node I
1 = H2, 2 = O2, 3 = hot gas
MIO(I) - number of I/O's associated with mass flow circuit I

NIO(I) - number of I/O's associated with volume I
MODIR(I,J) - flow direction of I/O J in mass flow circuit I
1 = inlet flow, -1 = outlet flow
IMFC(I,J) - node number of I/O J in mass flow circuit I
IODIR(I,J) - flow direction of I/O J in volume I
1 = inlet flow, -1 = outlet flow
IVOLN(I,J) - node number of I/O J in volume I
NODHG(I) - node number of hot gas flow I
NH2HG(I) - number of H2 flows entering hot gas flow I
NO2HG(I) - number of O2 flows entering hot gas flow I
ICEFF(I) - position in TTB array containing the combustion efficiency of hot gas flow I
IH2HG(I,J) - node number of H2 feed J to hot gas flow I
IO2HG(I,J) - node number of O2 feed J to hot gas flow I

Input File UDAT (uncertainty estimates data input file)
(file name variable identified in FILEIO.DAT)

IPRPD - unused in this version
ITTBD - unused in this version
IROWA - unused in this version
JCOLA - unused in this version
ITORDX - unused in this version
ITORDY - unused in this version
DPF - pressure fractional increment used in finite difference approximation of partial derivatives with respect to pressure
DTF - temperature fractional increment used in finite difference approximation of partial derivatives with respect to temperature
UP(I) - pressure uncertainty at node I
UT(I) - temperature uncertainty at node I
UW(I) - mass flow uncertainty at node I
UWMFC(I) - mass flow uncertainty associated with mass flow circuit I
UEVOL(I) - energy uncertainty associated with volume I
USVOL(I) - entropy uncertainty associated with volume I

Output Variable Definitions:

No output variables

Subroutine Calls:

RECON

Calling Routines:

None

Routine: RECON (reconciliation model construction and routing)

Routine Function:

Constructs a sequential quadratic programming (SQP) problem whose solution is the optimum reconciliation of uncertain test data and limited theoretical predictions for pressure, temperature and mass flow at specified node locations within the SSME flow network. Routes SQP solution logic. Outputs solution of SQP problem.

Common Blocks:

Blank	- quadratic programming algorithm matrices and parameters
TDAT	- test data
UDAT	- uncertainty estimates
VDAT	- control volume definition information
H2PRP	- hydrogen table pressures and temperatures, and associated enthalpies, entropies, and densities
O2PRP	- oxygen table pressures and temperatures, and associated enthalpies, entropies, and densities
H2OPRP	- water table pressures and temperatures, and associated enthalpies, entropies, and densities
TABLE	- number of distinct pressures and temperatures in each hydrogen, oxygen, and water table
STD	- reference state enthalpies entropies, and absolute entropies for hydrogen, oxygen, and water

Input Variable Definitions:

Common block inputs

Output Variable Definitions:

I	- node number
TTB(IP(I))	- original test pressure at node I
TTB(IT(I))	- original test temperature at node I
TTB(IW(I))	- original test mass flow rate at node I
PREC	- reconciled pressure at node I
TREC	- reconciled temperature at node I
WREC	- reconciled mass flow rate at node I
PADJ	- pressure adjustment made at node I
TADJ	- temperature adjustment made at node I
WADJ	- mass flow rate adjustment made at node I
PPCT	- percentage pressure adjustment made at node I
TPCT	- percentage temperature adjustment made at node I
WPCT	- percentage mass flow rate adjustment made at node I

Subroutine Calls:

PROP
CPIVOT

Calling Routines:

MAIN

Routine: CPIVOT (solver routing routine)

Routine Function:

The main routing routine for the complementary pivot method, quadratic programming problem solver.

Common Blocks:

Blank - quadratic programming algorithm matrices and parameters

Input Variable Definitions:

Common block inputs
N - dimension of square (NxN) main solver array
N = 6*number of nodes + 3*number of volumes
+ 2*number of mass flow circuits

Output Variable Definitions:

Common block outputs

Subroutine Calls:

MATRIX
INITIA
NEWBAS
SORT

Calling Routines:

RECON

Routine: MATRIX

Routine Function:

Initializes solver inputs

Common Blocks:

Blank - quadratic programming algorithm matrices and parameters

Input Variable Definitions:

Common block inputs

N - dimension of square (NxN) main solver array
N = 6*number of nodes + 3*number of volumes + 2*number of mass flow circuits

Output Variable Definitions:

Common block outputs

Subroutine Calls:

None

Calling Routines:

CPIVOT

Routine: INITIA

Routine Function:

Determines the initial almost complementary solution in the complementary pivot method solution strategy

Common Blocks:

Blank - quadratic programming algorithm matrices and parameters

Input Variable Definitions:

Common block inputs

N - dimension of square (NxN) main solver array
N = 6*number of nodes + 3*number of volumes
+ 2*number of mass flow circuits

Output Variable Definitions:

Common block outputs

Subroutine Calls:

None

Calling Routines:

CPIVOT

Routine: NEWBAS

Routine Function:

Finds the new basis column to enter in terms of the current basis in the complementary pivot method solver

Common Blocks:

Blank - quadratic programming algorithm matrices and parameters

Input Variable Definitions:

Common block inputs

N - dimension of square (NxN) main solver array
N = 6*number of nodes + 3*number of volumes
+ 2*number of mass flow circuits

Output Variable Definitions:

Common block outputs

Subroutine Calls:

SOLVE

Calling Routines:

CPIVOT

Routine: SORT

Routine Function:

Finds the pivot row for the next iteration by the use of (simplex-type) minimum ratio rule as part of the complementary pivot method solver

Common Blocks:

Blank - quadratic programming algorithm matrices and parameters

Input Variable Definitions:

Common block inputs

N - dimension of square (NxN) main solver array
N = 6*number of nodes + 3*number of volumes + 2*number of mass flow circuits

Output Variable Definitions:

Common block outputs

Subroutine Calls:

None

Calling Routines:

CPIVOT

Routine: PIVOT

Routine Function:

Performs the pivot operation by updating the inverse of the basis and the Q vector as part of the complementary pivot method solver

Common Blocks:

Blank - quadratic programming algorithm matrices and parameters

Input Variable Definitions:

Common block inputs

N - dimension of square (NxN) main solver array
N = 6*number of nodes + 3*number of volumes + 2*number of mass flow circuits

Output Variable Definitions:

Common block outputs

Subroutine Calls:

None

Calling Routines:

CPIVOT

Routine: SOLVE

Routine Function:

Correlates quadratic programming problem solution as part of the complementary pivot method solver

Common Blocks:

Blank - quadratic programming algorithm matrices and parameters

Input Variable Definitions:

Common block inputs

N - dimension of square (NxN) main solver array
N = 6*number of nodes + 3*number of volumes + 2*number of mass flow circuits

Output Variable Definitions:

Common block outputs

Subroutine Calls:

None

Calling Routines:

NEWBAS

Routine: PROP

Routine Function:

Calculates hydrogen, oxygen, and hot gas properties

Common Blocks:

H2PRP - hydrogen table pressures and temperatures, and associated enthalpies, entropies, and densities
O2PRP - oxygen table pressures and temperatures, and associated enthalpies, entropies, and densities
H2OPRP- water table pressures and temperatures, and associated enthalpies, entropies, and densities
TABLE - number of distinct pressures and temperatures in each hydrogen, oxygen, and water table

Input Variable Definitions:

Common block inputs
MAT - material type
1 = H2, 2 = O2, 3 = hot gas
PRSI - pressure
TMPI - temperature
OF - O2/H2 mass ratio
CEFF - combustion efficiency

Output Variable Definitions:

Common block outputs
ZENTH - table enthalpy
ZENTR - table entropy
ZDENS - density

Subroutine Calls:

PRPSAT
PRPMIX
ITERP2

Calling Routines:

RECON

Routine: PRPMIX

Routine Function:

Calculates hot gas mixture thermodynamic properties using a Dalton model

Common Blocks:

H2PRP - hydrogen table pressures and temperatures, and associated enthalpies, entropies, and densities
O2PRP - oxygen table pressures and temperatures, and associated enthalpies, entropies, and densities
H2OPRP- water table pressures and temperatures, and associated enthalpies, entropies, and densities
TABLE - number of distinct pressures and temperatures in each hydrogen, oxygen, and water table
STD - reference state enthalpies entropies, and absolute entropies for hydrogen, oxygen, and water

Input Variable Definitions:

Common block inputs
P - mixture pressure
TMPI - mixture temperature
OF - O2/H2 mass ratio
CEFF - combustion efficiency

Output Variable Definitions:

Common block outputs
HMIX - mixture enthalpy
SMIX - mixture entropy
PH2 - error "out of range" hydrogen pressure
PH2O - error "out of range" water pressure
PO2 - error "out of range" oxygen pressure
TMPI - error "out of range" temperature

Subroutine Calls:

ITERP2

Calling Routines:

PROP

Routine: PRPSAT

Routine Function:

Calculates thermodynamic properties near saturation curve

Common Blocks:

H2PRP - hydrogen table pressures and temperatures, and associated enthalpies, entropies, and densities
O2PRP - oxygen table pressures and temperatures, and associated enthalpies, entropies, and densities
H2OPRP- water table pressures and temperatures, and associated enthalpies, entropies, and densities
TABLE - number of distinct pressures and temperatures in each hydrogen, oxygen, and water table
STD - reference state enthalpies entropies, and absolute entropies for hydrogen, oxygen, and water

Input Variable Definitions:

Common block inputs
X - pressure
Y - temperature
TCRT - table critical temperature
NX1 - number of pressure values in table
NY1 - number of temperature values in table
NX2 -
YL - table low temperature
YH - table high temperature
PRS1 - pressure table values
TMP1 - temperature table values
PROP - thermodynamic property table values
PRS2 - saturation pressure table
TMP2 - saturation temperature table
PROPL - saturated liquid property value
PROPV - saturated vapor property value

Output Variable Definitions:

Common block outputs
FPROP - calculated thermodynamic property

Subroutine Calls:

ITERP2
ITERP1

Calling Routines:

PROP

APPENDIX C2

RECONCILER SOURCE CODE
VERSION RV2-0610

```
      PROGRAM RECONV2

      CHARACTER*24 DESC
      CHARACTER*12 PDAT,TDAT,UDAT,VDAT,ODAT,ODATV
C
C
      COMMON CPM(200,200), CPQ(200), L1CP, CPB(200,200), NL1CP, NL2CP,
     1       CPA(200), NE1CP, NE2CP, IRCP, MBASIS(300),
     2       CPW(200), CPZ(200)
C
      COMMON /VDAT/ IENV, IPCTTH, MAXSTG, NHG, NTMFC, NTNOD, NTVOL,
     1              IA(20), IP(20), IT(20), IW(20), MAT(20),
     2              MIO(5), MODIR(5,20), IMFCN(5,20),
     3              NIO(5), IODIR(5,20), IVOLN(5,20),
     4              NH2HG(5), NO2HG(5), NODHG(5), ICEFF(5),
     5              IH2HG(5,5), IO2HG(5,5)
C
      COMMON /TDAT/ TTB(100), NDESC, NTTB, DESC(5)
C
      COMMON /UDAT/ IPRPD, ITTBD, IROWA , JCOLA , ITORDX, ITORDY,
     1              DPF   , DTF   , UP(20), UT(20), UW(20),
     2              UEVOL(5), USVOL(5), UWMFC(5)
C
      COMMON /H2PRP/
     * H2P1(15),H2T1(11),H2H1(15,11),H2S1(15,11),H2D1(15,11),
     * H2P2(20),H2T2(11),H2H2(20,11),H2S2(20,11),H2D2(20,11),
     * H2P3(29),H2T3(25),H2H3(29,25),H2S3(29,25),H2D3(29,25),
     * H2P4(23),H2T4(25),H2H4(23,25),H2S4(23,25),H2D4(23,25)
      COMMON /O2PRP/
     * O2P1(13),O2T1(16),O2H1(13,16),O2S1(13,16),O2D1(13,16),
     * O2P2(13),O2T2(17),O2H2(13,17),O2S2(13,17),O2D2(13,17),
     * O2P3(5), O2T3(61),O2H3(5,61), O2S3(5,61), O2D3(5,61)
      COMMON /H2OPRP/
     * H2OP1(7),H2OT1(13),H2OH1(7,13),H2OS1(7,13),H2OD1(7,13)
C
      COMMON /TABLE/
     * NH2P(4),NH2T(4),NO2P(3),NO2T(3),NH2OP(1),NH2OT(1)
      COMMON /STD/
     * HH2REF,HO2REF,HWAREF,SH2REF,SO2REF,SWAREF,SH2A,SO2A,
     * SWAA
C
      DIMENSION
     * NH2PA(4),NH2TA(4),NO2PA(3),NO2TA(3),NH2OPA(1),NH2OTA(1)
C
      CHARACTER*70 PTITLE
C
      DATA (NH2PA(I),I=1,4)/15,20,29,23/
      DATA (NH2TA(J),J=1,4)/11,11,25,25/
      DATA (NO2PA(I),I=1,3)/13,13,5/
      DATA (NO2TA(J),J=1,3)/16,17,61/
      DATA (NH2OPA(I),I=1,1)/7/
      DATA (NH2OTA(J),J=1,1)/13/
C
      DO 90 I=1,4
      NH2P(I)=NH2PA(I)
   90 NH2T(I)=NH2TA(I)
      DO 91 I=1,3
      NO2P(I)=NO2PA(I)
   91 NO2T(I)=NO2TA(I)
      NH2OP(1)=NH2OPA(1)
```

```
      NH2OT(1)=NH2OTA(1)
C
      HH2REF =  1790.091
      HO2REF =   234.681
      HWAREF =  1339.990
      SH2REF =    15.440
      SO2REF =     1.530
      SWAREF =     2.294
      SH2A   =    15.481
      SO2A   =     1.531
      SWAA   =     0.928
C
      OPEN ( 7, FILE = 'FILEIO.DAT' , STATUS = 'OLD' )
      READ ( 7, * )  PDAT, TDAT, UDAT, VDAT, ODAT, ODATV
C
      OPEN (  8, FILE = PDAT , STATUS = 'OLD' )
      OPEN ( 12, FILE = TDAT , STATUS = 'OLD' )
      OPEN ( 13, FILE = UDAT , STATUS = 'OLD' )
      OPEN ( 14, FILE = VDAT , STATUS = 'OLD' )
C
      OPEN ( 21, FILE = ODAT  )
      OPEN ( 22, FILE = ODATV )
C
C     **  READ IN H2 PROPERTY TABLE INTO ARRAYS  **
C
      DO 10 ITBL=1,4
C
      READ(8,902) PTITLE
      DO 10 I=1,NH2P(ITBL)
      DO 10 J=1,NH2T(ITBL)
C
      IF(ITBL.EQ.1) READ(8,*) H2P1(I),H2T1(J),
     1   H2H1(I,J),H2S1(I,J),H2D1(I,J)
      IF(ITBL.EQ.2) READ(8,*) H2P2(I),H2T2(J),
     1   H2H2(I,J),H2S2(I,J),H2D2(I,J)
      IF(ITBL.EQ.3) READ(8,*) H2P3(I),H2T3(J),
     1   H2H3(I,J),H2S3(I,J),H2D3(I,J)
      IF(ITBL.EQ.4) READ(8,*) H2P4(I),H2T4(J),
     1   H2H4(I,J),H2S4(I,J),H2D4(I,J)
C
   10 CONTINUE
C
C     **  READ IN O2 PROPERTY TABLE INTO ARRAYS  **
C
      DO 20 ITBL=1,3
C
      READ(8,902) PTITLE
      DO 20 I=1,NO2P(ITBL)
      DO 20 J=1,NO2T(ITBL)
C
      IF(ITBL.EQ.1) READ(8,*) O2P1(I),O2T1(J),
     1   O2H1(I,J),O2S1(I,J),O2D1(I,J)
      IF(ITBL.EQ.2) READ(8,*) O2P2(I),O2T2(J),
     1   O2H2(I,J),O2S2(I,J),O2D2(I,J)
      IF(ITBL.EQ.3) READ(8,*) O2P3(I),O2T3(J),
     1   O2H3(I,J),O2S3(I,J),O2D3(I,J)
C
   20 CONTINUE
C
C     **  READ IN STEAM PROPERTY TABLES INTO ARRAYS  **
```

```
C
      DO 30  ITBL = 1, 1
C
      READ(8,902) PTITLE
      DO 30  I = 1, NH2OP(ITBL)
      DO 30  J = 1, NH2OT(ITBL)
C
      IF( ITBL .EQ. 1 )  READ(8,*) H2OP1(I),H2OT1(J),
     1   H2OH1(I,J),H2OS1(I,J),H2OD1(I,J)
C
   30 CONTINUE
C
      READ  (12,*)     NDESC, NTTB
      READ  (12,*)     ( DESC( I ),   I = 1, NDESC )
      READ  (12,*)     (   TTB( I ),  I = 1, NTTB   )
C
      WRITE (21,901)  ( DESC( I ),   I = 1, NDESC )
C
      READ (14,*)  IENV, IPCTTH, MAXSTG, NHG, NTMFC, NTNOD, NTVOL
      READ (14,*)  ( IA(I)  ,  I = 1, NTNOD ),
     2             ( IP(I)  ,  I = 1, NTNOD ),
     3             ( IT(I)  ,  I = 1, NTNOD ),
     4             ( IW(I)  ,  I = 1, NTNOD ),
     5             ( MAT(I),   I = 1, NTNOD ),
     6             ( MIO(I),   I = 1, NTMFC ),
     7             ( NIO(I),   I = 1, NTVOL )
C
      DO 50  I = 1, NTMFC
      READ (14,*)  ( MODIR(I,J),   J = 1, MIO(I) ),
     1             ( IMFCN(I,J),   J = 1, MIO(I) )
   50 CONTINUE
C
      DO 60  I = 1, NTVOL
      READ (14,*)  ( IODIR(I,J),   J = 1, NIO(I) ),
     1             ( IVOLN(I,J),   J = 1, NIO(I) )
   60 CONTINUE
C
      IF ( NHG .GT. 0 )  THEN
      READ (14,*)  ( NODHG(I),   I = 1, NHG ),
     1             ( NH2HG(I),   I = 1, NHG ),
     2             ( NO2HG(I),   I = 1, NHG ),
     3             ( ICEFF(I),   I = 1, NHG )
         DO 70  I = 1, NHG
         READ (14,*)  ( IH2HG(I,J),   J = 1, NH2HG(I) ),
     1                ( IO2HG(I,J),   J = 1, NO2HG(I) )
   70    CONTINUE
      ENDIF
C
      READ (13,*)  IPRPD, ITTBD, IROWA, JCOLA, ITORDX, ITORDY,
     1      DPF, DTF,
     2      ( UP(I),     I = 1, NTNOD ),
     3      ( UT(I),     I = 1, NTNOD ),
     4      ( UW(I),     I = 1, NTNOD ),
     5      ( UWMFC(I), I = 1, NTMFC ),
     6      ( UEVOL(I), I = 1, NTVOL ),
     7      ( USVOL(I), I = 1, NTVOL )
C
      CALL RECON
C
  901 FORMAT ( 10 ( /, 1X, A24 ) )
```

```
902   FORMAT ( /A70/ )
C
C
      STOP
      END
C**********************************************************************
      SUBROUTINE RECON
C
C     RECON - RECONCILIATION
C
      CHARACTER*24 DESC
      REAL JOULE
C
      DIMENSION   DDDPN(20), DDDTN(20), DHDPN(20), DHDTN(20),
     1            DSDPN(20), DSDTN(20), DENS(20) ,
     2            ASTD(20) , HSTD(20) , SSTD(20) ,
     3            REVA(20) , REVP(20) , REVT(20) , REVW(20) ,
     4            CPQQT(100,100), A(100,100), TTBREV(100)
C
      COMMON CPM(200,200), CPQ(200), L1CP, CPB(200,200), NL1CP, NL2CP,
     1       CPA(200), NE1CP, NE2CP, IRCP, MBASIS(300) ,
     2       CPW(200), CPZ(200)
C
      COMMON /VDAT/ IENV, IPCTTH, MAXSTG, NHG, NTMFC, NTNOD, NTVOL,
     1              IA(20), IP(20), IT(20), IW(20), MAT(20),
     2              MIO(5), MODIR(5,20), IMFCN(5,20),
     3              NIO(5), IODIR(5,20), IVOLN(5,20),
     4              NH2HG(5), NO2HG(5), NODHG(5), ICEFF(5),
     5              IH2HG(5,5), IO2HG(5,5)
C
      COMMON /TDAT/ TTB(100), NDESC, NTTB, DESC(5)
C
      COMMON /UDAT/ IPRPD, ITTBD, IROWA , JCOLA , ITORDX, ITORDY,
     1              DPF  , DTF  , UP(20), UT(20), UW(20),
     2              UEVOL(5), USVOL(5), UWMFC(5)
C
      COMMON /H2PRP/
     1 H2P1(15),H2T1(11),H2H1(15,11),H2S1(15,11),H2D1(15,11),
     2 H2P2(20),H2T2(11),H2H2(20,11),H2S2(20,11),H2D2(20,11),
     3 H2P3(29),H2T3(25),H2H3(29,25),H2S3(29,25),H2D3(29,25),
     4 H2P4(23),H2T4(25),H2H4(23,25),H2S4(23,25),H2D4(23,25)
      COMMON /O2PRP/
     1 O2P1(13),O2T1(16),O2H1(13,16),O2S1(13,16),O2D1(13,16),
     2 O2P2(13),O2T2(17),O2H2(13,17),O2S2(13,17),O2D2(13,17),
     3 O2P3(5), O2T3(61),O2H3(5,61), O2S3(5,61), O2D3(5,61)
      COMMON /H2OPRP/
     1 H2OP1(7),H2OT1(13),H2OH1(7,13),H2OS1(7,13),H2OD1(7,13)
C
      COMMON /STD/
     1 HH2REF,HO2REF,HWAREF,SH2REF,SO2REF,SWAREF,SH2A,SO2A,
     2 SWAA
      COMMON /TABLE/
     1 NH2P(4),NH2T(4),NO2P(3),NO2T(3),NH2OP(1),NH2OT(1)
C
      PARAMETER ( JOULE = 778.16, GC = 32.174 )
C
      ISTG  =  1
C
      TENV  =  TTB( IENV )
C
```

```
      DO 10  I = 1, NTNOD
      REVA(I)  =  TTB( IA(I) )
      REVP(I)  =  TTB( IP(I) )
      REVT(I)  =  TTB( IT(I) )
      REVW(I)  =  TTB( IW(I) )
   10 CONTINUE
C
   20 DO 70  I = 1, NTNOD
      P      =  REVP( I )
      T      =  REVT( I )
      W      =  REVW( I )
C
      IF ( MAT(I) .GE. 3 )  GO TO 40
      IF ( MAT(I) .GE. 2 )  GO TO 30
C
      CALL PROP ( 1, P, T, 0.0, 0.0, H, S, RHO)
      DENS(I)   =  RHO
      HN        =  H   - HH2REF
      HSTD(I)   =  HN
      SN        =  S   - SH2REF + SH2A
      SSTD(I)   =  SN
      AN        =  HN - TENV    * SN
      ASTD(I)   =  AN
C
      P2          =  P + DPF * P
      CALL PROP ( 1, P2, T, 0.0, 0.0, H2, S2, RHO2)
      HN2         =  H2  - HH2REF
      SN2         =  S2  - SH2REF + SH2A
      DDDPN(I)    =  ( RHO2 - RHO ) / ( P2 - P )
      DHDPN(I)    =  ( HN2 - HN )   / ( P2 - P )
      DSDPN(I)    =  ( SN2 - SN )   / ( P2 - P )
C
      T2          =  T + DTF * T
      CALL PROP ( 1, P, T2, 0.0, 0.0, H2, S2, RHO2)
      HN2         =  H2  - HH2REF
      SN2         =  S2  - SH2REF + SH2A
      DDDTN(I)    =  ( RHO2 - RHO ) / ( T2 - T )
      DHDTN(I)    =  ( HN2 - HN )   / ( T2 - T )
      DSDTN(I)    =  ( SN2 - SN )   / ( T2 - T )
C
      GO TO 70
C
   30 CALL PROP ( 2, P, T, 0.0, 0.0, H, S, RHO)
      DENS(I)   =  RHO
      HN        =  H   - HO2REF
      HSTD(I)   =  HN
      SN        =  S   - SO2REF + SO2A
      SSTD(I)   =  SN
      AN        =  HN - TENV    * SN
      ASTD(I)   =  AN
C
      P2          = P + DPF * P
      CALL PROP ( 2, P2, T, 0.0, 0.0, H2, S2, RHO2)
      HN2         =  H2  - HO2REF
      SN2         =  S2  - SO2REF + SO2A
      DDDPN(I)    =  ( RHO2 - RHO ) / ( P2 - P )
      DHDPN(I)    =  ( HN2 - HN )   / ( P2 - P )
      DSDPN(I)    =  ( SN2 - SN )   / ( P2 - P )
C
      T2          =  T + DTF * T
```

```
      CALL PROP ( 2, P, T2 ,0.0, 0.0, H2, S2, RHO2)
      HN2          =  H2  - HO2REF
      SN2          =  S2  - SO2REF + SO2A
      DDDTN(I)     =  ( RHO2 - RHO ) / ( T2 - T )
      DHDTN(I)     =  ( HN2 - HN )   / ( T2 - T )
      DSDTN(I)     =  ( SN2 - SN )   / ( T2 - T )
C
      GO TO 70
C
   40 DO 42  IDHG = 1, NHG
      IF ( NODHG(IDHG) .EQ. I )  THEN
         IHG  =  IDHG
         GO TO 44
      ELSE
      ENDIF
   42 CONTINUE
C
   44 WH2    =  0.0
      DO 50  IH2IN = 1, NH2HG( IHG )
      NNUM   =  IH2HG( IHG, IH2IN )
      WH2    =  WH2 + REVW( NNUM )
   50 CONTINUE
C
      WO2    =  0.0
      DO 60  IO2IN = 1, NO2HG( IHG )
      NNUM   =  IO2HG( IHG, IO2IN )
      WO2    =  WO2 + REVW( NNUM )
   60 CONTINUE
C
      CEFF       =  TTB( ICEFF( IHG ) )
      OF         =  WO2 / WH2
      CALL PROP ( 4, P, T, OF, CEFF, HMIX, SMIX, DMIX)
      DENS(I)    =  DMIX
      HSTD(I)    =  HMIX
      SSTD(I)    =  SMIX
      AMIX       =  HMIX - TENV * SMIX
      ASTD(I)    =  AMIX
C
      P2           =  P + DPF * P
      CALL PROP ( 4, P2, T, OF, CEFF, H2MIX, S2MIX, D2MIX)
      DDDPN(I)     =  ( D2MIX - DMIX ) / ( P2 - P )
      DHDPN(I)     =  ( H2MIX - HMIX ) / ( P2 - P )
      DSDPN(I)     =  ( S2MIX - SMIX ) / ( P2 - P )
C
      T2           =  T + DTF * T
      CALL PROP ( 4, P, T2, OF, CEFF, H2MIX, S2MIX, D2MIX)
      DDDTN(I)     =  ( D2MIX - DMIX ) / ( T2 - T )
      DHDTN(I)     =  ( H2MIX - HMIX ) / ( T2 - T )
      DSDTN(I)     =  ( S2MIX - SMIX ) / ( T2 - T )
C
   70 CONTINUE
C
      M     =  3 * NTNOD + 2 * NTMFC + 3 * NTVOL
      N     =  3 * NTNOD
C
      DO 80  I = 1, M
      DO 80  J = 1, N
      A(I,J)   =  0.0
   80 CONTINUE
C
```

```
      DO 82   I = 1, N
      DO 82   J = 1, N
      CPQQT(I,J)  =   0.0
   82 CONTINUE
C
      DO 84   ITNOD = 1, NTNOD
      I1                 =  ITNOD
      I2                 =  ITNOD   + NTNOD
      I3                 =  ITNOD   + 2 * NTNOD
      CPQQT(I1,I1)       =  4. / UW( ITNOD ) ** 2
      CPQQT(I2,I2)       =  4. / UP( ITNOD ) ** 2
      CPQQT(I3,I3)       =  4. / UT( ITNOD ) ** 2
      CPQ(I1)            = -4. / UW( ITNOD )
      CPQ(I2)            = -4. / UP( ITNOD )
      CPQ(I3)            = -4. / UT( ITNOD )
   84 CONTINUE
C
      DO 90   ITNOD = 1, NTNOD
      I1           =  ITNOD
      I2           =  ITNOD + NTNOD
      I3           =  ITNOD + 2 * NTNOD
      A(I1,I1)     =  -1.
      A(I2,I2)     =  -1.
      A(I3,I3)     =  -1.
      CPQ(N+I1)    =   2. * UW( I1 )
      CPQ(N+I2)    =   2. * UP( I1 )
      CPQ(N+I3)    =   2. * UT( I1 )
   90 CONTINUE
C
      NQROW   =   2 * N
C
      DO 98   ITMFC = 1, NTMFC
      SUMM    =  0.
      I1      =  ITMFC
      I2      =  ITMFC + NTMFC
C
      DO 92   IIO = 1, MIO( ITMFC )
      IOD     =  MODIR( ITMFC, IIO )
      IMN     =  IMFCN( ITMFC, IIO )
      W       =  REVW( IMN )
      SUMM    =  SUMM + IOD * ( W - UW( IMN ) )
   92 CONTINUE
C
      DO 94   IIO = 1, MIO( ITMFC )
      IOD              =    MODIR( ITMFC, IIO )
      IMN              =    IMFCN( ITMFC, IIO )
      A(N+I1,IMN)      =    IOD
      A(N+I2,IMN)      =  - IOD
   94 CONTINUE
C
      CPQ(NQROW+I1)   =     SUMM + UWMFC( ITMFC )
      CPQ(NQROW+I2)   =   - SUMM + UWMFC( ITMFC )
   98 CONTINUE
C
      NAROW   =   N + 2 * NTMFC
      NQROW   =   2 * N + 2 * NTMFC
C
      DO 110   ITVOL = 1, NTVOL
      I1      =  ITVOL
      I2      =  ITVOL + NTVOL
```

```
      I3      =  ITVOL + 2 * NTVOL
      CEO     =  0.
      CSO     =  0.
      SUMMI   =  0.
C
      DO 99  IIO = 1, NIO( ITVOL )
      IOD    =  IODIR( ITVOL, IIO )
      IVN    =  IVOLN( ITVOL, IIO )
      AREA   =  REVA( IVN )
      W      =  REVW( IVN )
      RHO    =  DENS( IVN )
      ENTH   =  HSTD( IVN )
      ENTR   =  SSTD( IVN )
      AVAIL  =  ASTD( IVN )
C
      CEO1   =  IOD * W * ENTH
      IF ( AREA .GT. 0. ) THEN
         CEO2  =  IOD * W**3 / ( 2. * GC * JOULE * RHO**2 * AREA**2 )
      ELSE
         CEO2  =  0.
      ENDIF
      CEO    =  CEO + CEO1 + CEO2
      CSO    =  CSO + IOD * W * ENTR
   99 CONTINUE
C
      SUMEQ  =  0.0
      SUMSQ  =  0.0
C
      DO 100  IIO = 1, NIO( ITVOL )
      IOD    =  IODIR( ITVOL, IIO )
      IVN    =  IVOLN( ITVOL, IIO )
      AREA   =  REVA( IVN )
      W      =  REVW( IVN )
      P      =  REVP( IVN )
      T      =  REVT( IVN )
      RHO    =  DENS( IVN )
      ENTH   =  HSTD( IVN )
      ENTR   =  SSTD( IVN )
      AVAIL  =  ASTD( IVN )
      DDDP   =  DDDPN( IVN )
      DHDP   =  DHDPN( IVN )
      DSDP   =  DSDPN( IVN )
      DDDT   =  DDDTN( IVN )
      DHDT   =  DHDTN( IVN )
      DSDT   =  DSDTN( IVN )
C
      CEM  =  IOD * ENTH
      CEP  =  IOD * W * DHDP
      CET  =  IOD * W * DHDT
      CSM  =  IOD * ENTR
      CSP  =  IOD * W * DSDP
      CST  =  IOD * W * DSDT
C
      IF ( AREA .GT. 0. )  THEN
         CEM   =  CEM + 3. * IOD * W**2 /
     1            ( 2. * GC * JOULE * RHO**2 * AREA**2 )
         CEP   =  CEP - IOD * W**3 * DDDP /
     1            ( GC * JOULE * RHO**3 * AREA**2 )
         CET   =  CET - IOD * W**3 * DDDT /
     1            ( GC * JOULE * RHO**3 * AREA**2 )
```

```
      ENDIF
C
      NROW1                      =   NAROW + I1
      A(NROW1,IVN)               =   CEM
      A(NROW1,IVN+NTNOD)         =   CEP
      A(NROW1,IVN+2*NTNOD)       =   CET
C
      NROW2                      =   NAROW + I2
      A(NROW2,IVN)               =  -CEM
      A(NROW2,IVN+NTNOD)         =  -CEP
      A(NROW2,IVN+2*NTNOD)       =  -CET
C
      NROW3                      =   NAROW+I3
      A(NROW3,IVN)               =  -CSM
      A(NROW3,IVN+NTNOD)         =  -CSP
      A(NROW3,IVN+2*NTNOD)       =  -CST
C
      SUMEQ  =  SUMEQ + CEM * UW( IVN ) + CEP * UP( IVN ) +
     1                  CET * UT( IVN )
      SUMSQ  =  SUMSQ + CSM * UW( IVN ) + CSP * UP( IVN ) +
     1                  CST * UT( IVN )
C
 100  CONTINUE
C
      CPQ(NQROW+I1)  =    CEO + UEVOL( ITVOL ) - SUMEQ
      CPQ(NQROW+I2)  =  - CEO + UEVOL( ITVOL ) + SUMEQ
      CPQ(NQROW+I3)  =  - CSO + USVOL( ITVOL ) + SUMSQ
C
 110  CONTINUE
C
      DO 120  I = 1, N
      DO 120  J = 1, N
      CPM(I,J)   =   CPQQT( I, J )
 120  CONTINUE
C
      DO 130  I = 1, M
      DO 130  J = 1, N
      CPM(N+I,J)   =   A( I, J )
      CPM(J,N+I)   =  - A( I, J )
 130  CONTINUE
C
      NP1   =   N + 1
      MN    =   M + N
      DO 140  I = NP1, MN
      DO 140  J = NP1, MN
      CPM(I,J)   =   0.0
 140  CONTINUE
C
      CALL CPIVOT ( MN )
      IF ( IRCP .EQ. -1 )  WRITE ( 21, 986 )
      IF ( IRCP .EQ. -2 )  WRITE ( 21, 987 )
      IF ( IRCP .EQ. -3 )  WRITE ( 21, 988 )
C
      WRITE ( 21, 981 )
      WRITE ( 21, 982 )
C
      DO 300  I = 1, NTNOD
      WRITE ( 21, 951 )  I, TTB(IP(I)), TTB(IT(I)), TTB(IW(I))
 300  CONTINUE
C
```

```
      WRITE ( 21, 983 )
      WRITE ( 21, 982 )
C
      DO 310  I = 1, NTNOD
      REVW( I )  =  REVW( I ) + ( CPZ( I ) - UW( I ) )
      REVP( I )  =  REVP( I ) + ( CPZ( I + NTNOD ) - UP( I ) )
      REVT( I )  =  REVT( I ) + ( CPZ( I + 2 * NTNOD ) - UT( I ) )
      WREC  =  REVW( I )
      PREC  =  REVP( I )
      TREC  =  REVT( I )
      WRITE ( 21, 951 )  I, PREC, TREC, WREC
 310  CONTINUE
C
      WRITE ( 21, 984 )
      WRITE ( 21, 982 )
C
      DO 320  I = 1, NTNOD
      WADJ  =  REVW( I ) - TTB( IW( I ) )
      PADJ  =  REVP( I ) - TTB( IP( I ) )
      TADJ  =  REVT( I ) - TTB( IT( I ) )
      WRITE ( 21, 951 )  I, PADJ, TADJ, WADJ
 320  CONTINUE
C
      WRITE ( 21, 985 )
      WRITE ( 21, 982 )
C
      DO 330  I = 1, NTNOD
      WPCT  =  100 * ( REVW( I ) - TTB( IW( I ) ) ) / TTB( IW( I ) )
      PPCT  =  100 * ( REVP( I ) - TTB( IP( I ) ) ) / TTB( IP( I ) )
      TPCT  =  100 * ( REVT( I ) - TTB( IT( I ) ) ) / TTB( IT( I ) )
      WRITE ( 21, 951 )  I, PPCT, TPCT, WPCT
 330  CONTINUE
C
      IF ( ISTG .LT. MAXSTG ) THEN
         ISTG  =  ISTG + 1
         GO TO 20
      ELSE
      ENDIF
C
 500  CONTINUE
C
      TTBREV( IENV )    =  TTB( IENV )
      TTBREV( IPCTTH ) =  TTB( IPCTTH )
C
      DO 510  I = 1, NTNOD
      TTBREV( IA(I) )  =  REVA( I )
      TTBREV( IP(I) )  =  REVP( I )
      TTBREV( IT(I) )  =  REVT( I )
      TTBREV( IW(I) )  =  REVW( I )
 510  CONTINUE
C
      DO 520  I = 1, NHG
      TTBREV( ICEFF(I) )  =  TTB( ICEFF(I) )
 520  CONTINUE
C
      WRITE ( 22, *   )  NDESC, NTTB
      WRITE ( 22, 990 )  ( DESC(I)   ,  I = 1, NDESC )
      WRITE ( 22, 989 )  ( TTBREV(I),  I = 1, NTTB )
C
 951  FORMAT ( 9X, I6, 3F15.2 )
```

```
981   FORMAT ( //2X, 'ORIGINAL NODE DATA' )
982   FORMAT ( /11X, 'NODE', 11X, 'PRES', 11X, 'TEMP', 11X, 'FLOW')
983   FORMAT ( //2X, 'RECONCILED NODE DATA' )
984   FORMAT ( //2X, 'BALANCING ADJUSTMENTS' )
985   FORMAT ( //2X, 'PERCENT ADJUSTMENT' )
986   FORMAT ( //2X, 'NO COMPLEMENTARY SOLUTION OBTAINED' )
987   FORMAT ( //2X, 'TRIVIAL COMPLEMENTARY SOLUTION' )
988   FORMAT ( //2X, 'SOLUTION OBTAINED' )
989   FORMAT ( 6F12.3 )
990   FORMAT ( 2X,1H',A24,1H' )
C
      RETURN
      END
C**********************************************************************
C
C
      SUBROUTINE CPIVOT (N)
C
      COMMON AM(200,200),Q(200),L1,B(200,200),NL1,NL2,A(200),NE1,NE2,
     1       IR,MBASIS(300),W(200),Z(200)
C
C DESCRIPITION OF PARAMETERS IN COMMON
C    AM        A TWO DIMENSIONAL ARRAY CONTAINING THE
C              ELEMENTS OF MATRIX M.
C    Q         A SINGLY SUBSCRIPTED ARRAY CONTAINING THE
C              ELEMENTS OF VECTOR Q.
C    L1        AN INTEGER VARIABLE INDICATING THE NUMBER OF
C              ITERATIONS TAKEN FOR EACH PROBLEM.
C    B         A TWO DIMENSIONAL ARRAY CONTAINING THE
C              ELEMENTS OF THE INVERSE OF THE CURRENT BASIS.
C    W         A SINGLY SUBSCRIPTED ARRAY CONTAINING THE VALUES
C              OF W VARIABLES IN EACH SOLUTION.
C    Z         A SINGLY SUBSCRIPTED ARRAY CONTAINING THE VALUES
C              OF Z VARIABLES IN EACH SOLUTION.
C    NL1       AN INTEGER VARIABLE TAKING VALUE 1 OR 2 DEPEND-
C              ING ON WHETHER VARIABLE W OR Z LEAVES THE BASIS
C    NE1       SIMILAR TO NL1 BUT INDICATES VARIABLE ENTERING
C    NL2       AN INTEGER VARIABLE INDICATING WHAT COMPONENT
C              OF W OR Z VARIABLE LEAVES THE BASIS.
C    NE2       SIMILAR TO NL2 BUT INDICATES VARIABLE ENTERING
C    A         A SINGLY SUBSCRIPTED ARRAY CONTAINING THE
C              ELEMENTS OF THE TRANSFPORMED COLUMN THAT IS
C              ENTERING THE BASIS.
C    IR        AN INTEGER VARIABLE DENOTING THE PIVOT ROW AT
C              EACH ITERATION. ALSO USED TO INDICATE
C              ALGORITHM TERMINATION
C              IR = -2   COMPLEMENTARY SOLUTION DETERMINED
C              IR = -1   PROBLEM HAS NO COMPLEMENTARY SOLUTION
C    MBASIS    A SINGLY SUBSCRIPTED ARRAY-INDICATOR FOR THE
C              BASIC VARIABLES. TWO INDICATORS ARE USED FOR
C              EACH BASIC VARIABLE-ONE INDICATING WHETHER
C              IT IS A W OR Z AND ANOTHER INDICATING WHAT
C              COMPONENT OF W OR Z.
C
C PROGRAM CALLING SEQUENCE, N IS THE SIZE OF MATRIX AM
C
C
      CALL MATRIX ( N )
      CALL INITIA ( N )
      IF ( IR .EQ. -2 )  GO TO 5
```

```
C
    4 CALL NEWBAS ( N )
      IF ( IR .EQ. -3 ) GO TO 5
C
      CALL SORT   ( N )
      IF ( IR .EQ. -1 )  GO TO 5
C
      CALL PIVOT  ( N )
      GO TO 4
C
    5 RETURN
      END
C
C
C
      SUBROUTINE MATRIX ( N )
C
C PURPOSE - TO INITIALIZE THE VARIOUS INPUT DATA
C
      COMMON AM(200,200), Q(200), L1, B(200,200), NL1, NL2, A(200),
     1       NE1, NE2, IR, MBASIS(300), W(200), Z(200)
C
C IN ITERATION 1, BASIS INVERSE IS AN IDENTITY MATRIX
C
      DO 5  J = 1, N
      DO 4  I = 1, N
      IF ( I .EQ. J )  GO TO 3
      B(I,J)  =  0.0
      GO TO 4
    3 B(I,J)=1.0
    4 CONTINUE
    5 CONTINUE
C
      RETURN
      END
C
C
C
      SUBROUTINE INITIA ( N )
C
C PURPOSE - TO FIND THE INITIAL ALMOST COMPLEMENTARY SOLUTION
C           BY ADDING AN ARTIFICIAL VARIABLE Z0.
C
      COMMON AM(200,200), Q(200), L1, B(200,200), NL1, NL2, A(200),
     1       NE1, NE2, IR, MBASIS(300), W(200), Z(200)
C
C SET Z0 EQUAL TO THE MOST NEGATIVE Q(I)
      I  =  1
      J  =  2
    1 IF ( Q(I) .LE. Q(J) )  GO TO 2
      I  =  J
    2 J  =  J + 1
      IF ( J .LE. N )  GO TO 1
C
C UPDATE Q VECTOR
      IR      =  I
      T1      = -Q( IR )
      IF ( T1 .LE. 0.0 )  GO TO 9
      DO 3  I = 1, N
      Q(I)    =  Q( I ) + T1
```

```
    3 CONTINUE
      Q(IR)    =   T1
C
C UPDATE BASIS INVERSE AND INDICATOR VECTOR
C OF BASIC VARIABLES
      DO 4  J = 1, N
      B(J,IR)     =   -1.0
      W(J)        =   Q( J )
      Z(J)        =   0.0
      MBASIS(J)   =   1
      L           =   N + J
      MBASIS(L)   =   J
    4 CONTINUE
C
      NL1          =  1
      L            =  N + IR
      NL2          =  IR
      MBASIS(IR)   =  3
      MBASIS(L)    =  0
      W(IR)        =  0.0
      Z0           =  Q( IR )
      L1           =  1
C
      RETURN
C
    9 IR  =  -2
      RETURN
      END
C
C
C
      SUBROUTINE NEWBAS ( N )
C
C PURPOSE - TO FIND THE NEW BASIS COLUMN TO ENTER IN
C           TERMS OF THE CURRENT BASIS.
C
      COMMON AM(200,200), Q(200), L1, B(200,200), NL1, NL2, A(200),
     1       NE1, NE2, IR, MBASIS(300), W(200), Z(200)
C
C IF NL1 IS NEITHER 1 NOR 2 THEN THE VARIABLE Z0 LEAVES THE
C BASIS INDICATING TERMINATION WITH A COMPLEMENTARY SOLUTION
      IF ( NL1 .EQ. 1 )  GO TO 2
      IF ( NL1 .EQ. 2 )  GO TO 5
C
      CALL SOLVE ( N )
      IR   =  -3
      RETURN
C
    2 NE1  =  2
      NE2  =  NL2
C
C UPDATE NEW BASIC COLUMN BY MULTIPLYING BY BASIS INVERSE.
      DO 4  I = 1, N
      T1    =  0.0
      DO 3  J = 1, N
    3 T1    =  T1 - B( I, J ) * AM( J, NE2 )
      A(I)  =  T1
    4 CONTINUE
      RETURN
C
```

```
    5 NE1    =  1
      NE2    =  NL2
      DO 6   I = 1, N
      A(I)   =  B( I, NE2 )
    6 CONTINUE
C
      RETURN
      END
C
C
C
      SUBROUTINE SORT ( N )
C
C PURPOSE - TO FIND THE PIVOT ROW FOR THE NEXT ITERATION BY THE
C           USE OF (SIMPLEX-TYPE) MINIMUM RATIO RULE.
C
      COMMON AM(200,200), Q(200), L1, B(200,200), NL1, NL2, A(200),
     1       NE1, NE2, IR, MBASIS(300), W(200), Z(200)
C
      I   =  1
    1 IF ( A( I ) .GT. 0.0 )  GO TO 2
      I   =  I + 1
      IF ( I .GT. N )  GO TO 6
      GO TO 1
C
    2 T1  =  Q( I ) / A( I )
      IR  =  I
    3 I   =  I + 1
      IF ( I .GT. N )  GO TO 5
      IF ( A( I ) .GT. 0.0 )  GO TO 4
      GO TO 3
C
    4 T2  =  Q( I ) / A( I )
      IF ( T2 .GE. T1 )  GO TO 3
      IR  =  I
      T1  =  T2
      GO TO 3
C
    5 RETURN
C
C FAILURE OF THE RATIO RULE INDICATES TERMINATION WITH
C NO COMPLEMENTARY SOLUTION.
C
    6 IR  =  -1
      RETURN
      END
C
C
C
      SUBROUTINE PIVOT ( N )
C
C PURPOSE - TO PERFORM THE PIVOT OPERATION BY UPDATING THE
C           THE INVERSE OF THE BASIS AND Q VECTOR.
C
      COMMON AM(200,200), Q(200), L1, B(200,200), NL1, NL2, A(200),
     1       NE1, NE2, IR, MBASIS(300), W(200), Z(200)
C
      DO 1  I = 1, N
    1 B(IR,I) =  B( IR, I ) / A( IR )
      Q(IR)   =  Q( IR ) / A( IR )
```

```
      DO 3  I = 1, N
      IF ( I .EQ. IR )  GO TO 3
      Q(I)     =  Q( I ) - Q( IR ) * A( I )
      DO 2  J = 1, N
      B(I,J)  =  B( I, J ) - B( IR, J ) * A( I )
    2 CONTINUE
    3 CONTINUE
C
C UPDATE THE INDICATOR VECTOR OF BASIC VARIABLES
      NL1           =  MBASIS( IR )
      L             =  N + IR
      NL2           =  MBASIS( L )
      MBASIS(IR)    =  NE1
      MBASIS(L)     =  NE2
      L1            =  L1 + 1
C
      RETURN
      END
C
C
C
      SUBROUTINE SOLVE ( N )
C
C PURPOSE - TO CORRELATE COMPLEMENTARY PROBLEM SOLUTION
C
      COMMON AM(200,200), Q(200), L1, B(200,200), NL1, NL2, A(200),
     1       NE1, NE2, IR, MBASIS(300), W(200), Z(200)
C
      DO 1  I = 1, N
      W(I)    =  0.0
      Z(I)    =  0.0
    1 CONTINUE
C
      I       =  N + 1
      J       =  1
C
    2 K1      =  MBASIS( I )
      K2      =  MBASIS( J )
C
      IF ( Q( J ) .GE. 0.0 )  GO TO 3
      Q(J)    =  0.0
C
    3 IF ( K2 .EQ. 1 )  GO TO 5
      Z(K1)   =  Q( J )
      GO TO 7
C
    5 W(K1)   =  Q( J )
    7 I       =  I + 1
      J       =  J + 1
      IF ( J .LE. N )  GO TO 2
C
      RETURN
      END
C

C
C**********************************************************************
      SUBROUTINE PROP (MAT,PRSI,TMPI,OF,CEFF,ZENTH,ZENTR,ZDENS)
C
C     PROP - PROPERTY PROGRAM CALCULATING HYDROGEN,
```

```
C                OXYGEN, STEAM AND HOT GAS PROPERTIES
C
      COMMON /H2PRP/
     * H2P1(15),H2T1(11),H2H1(15,11),H2S1(15,11),H2D1(15,11),
     * H2P2(20),H2T2(11),H2H2(20,11),H2S2(20,11),H2D2(20,11),
     * H2P3(29),H2T3(25),H2H3(29,25),H2S3(29,25),H2D3(29,25),
     * H2P4(23),H2T4(25),H2H4(23,25),H2S4(23,25),H2D4(23,25)
      COMMON /O2PRP/
     * O2P1(13),O2T1(16),O2H1(13,16),O2S1(13,16),O2D1(13,16),
     * O2P2(13),O2T2(17),O2H2(13,17),O2S2(13,17),O2D2(13,17),
     * O2P3(5), O2T3(61),O2H3(5,61), O2S3(5,61), O2D3(5,61)
      COMMON /H2OPRP/
     * H2OP1(7),H2OT1(13),H2OH1(7,13),H2OS1(7,13),H2OD1(7,13)
C
      COMMON /TABLE/
     * NH2P(4),NH2T(4),NO2P(3),NO2T(3),NH2OP(1),NH2OT(1)
C
      DIMENSION
     * TSH2(11),PSH2(11),HLH2(11),HVH2(11),SLH2(11),SVH2(11),
     * DLH2(11),DVH2(11),
     * TSO2(16),PSO2(16),HLO2(16),HVO2(16),SLO2(16),SVO2(16),
     * DLO2(16),DVO2(16)
C
C     TSH2  -  H2 SATURATION TEMPERATURE
C     PSH2  -  H2 SATURATION PRESSURE
C     HLH2  -  H2 SATURATION ENTHALPY - LIQUID
C     HVH2  -  H2 SATURATION ENTHALPY - VAPOR
C     SLH2  -  H2 SATURATION ENTROPY  - LIQUID
C     SVH2  -  H2 SATURATION ENTROPY  - VAPOR
C     DLH2  -  H2 SATURATION DENSITY  - LIQUID
C     DVH2  -  H2 SATURATION DENSITY  - VAPOR
C
          DATA (TSH2(J),J=1,11)/
     * 30.0,32.0,34.0,36.0,38.0,40.0,42.0,44.0,46.0,48.0,50.0/
C
          DATA (PSH2(J),J=1,11)/
     * 4.170,6.446,9.527,13.561,18.694,25.089,32.915,42.334,
     * 53.514,66.625,81.838/
C
          DATA (HLH2(J),J=1,11)/
     * -123.995,-120.090,-115.893,-111.380,-106.524,-101.289,
     * -95.636,-89.513,-82.850,-75.556,-67.493/
C
          DATA (HVH2(J),J=1,11)/
     * 70.977,74.584,77.848,80.729,83.256,85.199,86.614,87.431,
     * 87.546,86.817,85.043/
C
          DATA (SLH2(J),J=1,11)/
     * 1.506,1.629,1.752,1.876,2.002,2.129,2.259,2.391,2.528,
     * 2.670,2.819/
C
          DATA (SVH2(J),J=1,11)/
     * 8.005,7.713,7.451,7.214,6.998,6.794,6.601,6.415,6.234,
     * 6.054,5.871/
C
          DATA (DLH2(J),J=1,11)/
     * 4.6500,4.5832,4.5127,4.4378,4.3580,4.2724,4.1801,4.0798,
     * 3.9698,3.8479,3.7108/
C
          DATA (DVH2(J),J=1,11)/
```

```
     * 0.0272,0.0401,0.0568,0.0779,0.1039,0.1363,0.1757,0.2234,
     * 0.2809,0.3508,0.4362/
C
C     TSO2  -  O2 SATURATION TEMPERATURE
C     PSO2  -  O2 SATURATION PRESSURE
C     HLO2  -  O2 SATURATION ENTHALPY - LIQUID
C     HVO2  -  O2 SATURATION ENTHALPY - VAPOR
C     SLO2  -  O2 SATURATION ENTROPY  - LIQUID
C     SVO2  -  O2 SATURATION ENTROPY  - VAPOR
C     DLO2  -  O2 SATURATION DENSITY  - LIQUID
C     DVO2  -  O2 SATURATION DENSITY  - VAPOR
C
          DATA (TSO2(J),J=1,16)/
     * 160.0,164.0,168.0,172.0,176.0,180.0,184.0,188.0,192.0,
     * 196.0,200.0,204.0,208.0,212.0,216.0,220.0/
C
          DATA (PSO2(J),J=1,16)/
     * 12.810,16.183,20.200,24.935,30.467,36.876,44.243,52.654,
     * 62.194,72.951,85.013,98.473,113.421,129.952,148.162,
     * 168.146/
C
          DATA (HLO2(J),J=1,16)/
     * -58.356,-56.730,-55.096,-53.455,-51.804,-50.144,-48.473,
     * -46.790,-45.093,-43.380,-41.650,-39.901,-38.130,-36.334,
     * -34.511,-32.657/
C
          DATA (HVO2(J),J=1,16)/
     * 33.777,34.457,35.110,35.734,36.326,36.884,37.408,37.894,
     * 38.340,38.745,39.105,39.419,39.683,39.894,40.049,40.144/
C
          DATA (SLO2(J),J=1,16)/
     * 0.698,0.708,0.717,0.727,0.736,0.746,0.755,0.764,0.772,
     * 0.781,0.790,0.798,0.806,0.815,0.823,0.831/
C
          DATA (SVO2(J),J=1,16)/
     * 1.273,1.263,1.254,1.245,1.237,1.229,1.221,1.214,1.207,
     * 1.200,1.193,1.187,1.180,1.174,1.168,1.162/
C
          DATA (DLO2(J),J=1,16)/
     * 71.630,70.941,70.243,69.536,68.818,68.089,67.347,66.593,
     * 65.823,65.037,64.234,63.412,62.567,61.699,60.804,59.880/
C
          DATA (DVO2(J),J=1,16)/
     * 0.246,0.305,0.374,0.455,0.547,0.653,0.774,0.911,1.065,
     * 1.239,1.433,1.650,1.893,2.162,2.461,2.794/
C
C
   51  FORMAT(/3X,'PROP - REQUESTED PRS > ',F7.2,2X,
     *   'AND TMP > ',F7.2,2X,'FOR H2 IS OUT OF RANGE')
   52  FORMAT(/3X,'PROP - REQUESTED PRS > ',F7.2,2X,
     *   'AND TMP > ',F7.2,2X,'FOR O2 IS OUT OF RANGE')
   53  FORMAT(/3X,'PROP - REQUESTED PRS > ',F7.2,2X,
     *   'AND TMP > ',F7.2,2X,'FOR STEAM IS OUT OF RANGE')
C
C      **  INTERPOLATE RESULTS FROM SINGLE ARRAY  **
C
       IPRP=0
       NPX1=2
       NPY1=2
       ZENTH=0.0
```

```
      ZENTR=0.0
      ZDENS=0.0
C
      GO TO (10,20,30,40) MAT
C
   10 IF(TMPI.GT.   30.0.AND.TMPI.LT.  50.0) IPRP=1
      IF(TMPI.GT.   70.0.AND.TMPI.LT. 110.0) IPRP=2
      IF(TMPI.GT.  240.0.AND.TMPI.LT. 720.0) IPRP=3
      IF(TMPI.GT.1400.0.AND.TMPI.LT.2000.0) IPRP=4
      GO TO (11,12,13,14) IPRP
C
   11 IF(PRSI.LT.   20.0.OR.PRSI.GT. 370.0) GO TO 50
      CALL PRPSAT(PRSI,TMPI,ZENTH,
     *  TSH2(11),NH2P(1),NH2T(1),11,29.95,50.05,
     *  H2P1,H2T1,H2H1,PSH2,TSH2,HLH2,HVH2)
      CALL PRPSAT(PRSI,TMPI,ZENTR,
     *  TSH2(11),NH2P(1),NH2T(1),11,29.95,50.05,
     *  H2P1,H2T1,H2S1,PSH2,TSH2,SLH2,SVH2)
      CALL PRPSAT(PRSI,TMPI,ZDENS,
     *  TSH2(11),NH2P(1),NH2T(1),11,29.95,50.05,
     *  H2P1,H2T1,H2D1,PSH2,TSH2,DLH2,DVH2)
      RETURN
C
   12 IF(PRSI.LT.3400.0.OR.PRSI.GT.7200.0) GO TO 50
      CALL ITERP2(PRSI,TMPI,H2P2,H2T2,H2H2,
     *  NH2P(2),NH2T(2),NPX1,NPY1,NH2P(2),ZENTH,N1)
      CALL ITERP2(PRSI,TMPI,H2P2,H2T2,H2S2,
     *  NH2P(2),NH2T(2),NPX1,NPY1,NH2P(2),ZENTR,N1)
      CALL ITERP2(PRSI,TMPI,H2P2,H2T2,H2D2,
     *  NH2P(2),NH2T(2),NPX1,NPY1,NH2P(2),ZDENS,N1)
      RETURN
C
   13 IF(PRSI.LT.1400.0.OR.PRSI.GT.7000.0) GO TO 50
      CALL ITERP2(PRSI,TMPI,H2P3,H2T3,H2H3,
     *  NH2P(3),NH2T(3),NPX1,NPY1,NH2P(3),ZENTH,N1)
      CALL ITERP2(PRSI,TMPI,H2P3,H2T3,H2S3,
     *  NH2P(3),NH2T(3),NPX1,NPY1,NH2P(3),ZENTR,N1)
      CALL ITERP2(PRSI,TMPI,H2P3,H2T3,H2D3,
     *  NH2P(3),NH2T(3),NPX1,NPY1,NH2P(3),ZDENS,N1)
      RETURN
C
   14 IF(PRSI.LT.1400.0.OR.PRSI.GT.5800.0) GO TO 50
      CALL ITERP2(PRSI,TMPI,H2P4,H2T4,H2H4,
     *  NH2P(4),NH2T(4),NPX1,NPY1,NH2P(4),ZENTH,N1)
      CALL ITERP2(PRSI,TMPI,H2P4,H2T4,H2S4,
     *  NH2P(4),NH2T(4),NPX1,NPY1,NH2P(4),ZENTR,N1)
      CALL ITERP2(PRSI,TMPI,H2P4,H2T4,H2D4,
     *  NH2P(4),NH2T(4),NPX1,NPY1,NH2P(4),ZDENS,N1)
      RETURN
C
C
   20 IF(TMPI.GT. 160.0.AND.TMPI.LT. 240.0) IPRP=1
      IF(IPRP.EQ.1.AND.PRSI.LT.650.0) IPRP=1
      IF(IPRP.EQ.1.AND.PRSI.GT.650.0) IPRP=2
      IF(TMPI.GT. 600.0.AND.TMPI.LT.1500.0) IPRP=3
      GO TO (21,22,23) IPRP
C
   21 IF(PRSI.LT.   30.0.OR.PRSI.GT. 630.0) GO TO 50
      IF(TMPI.LT. 160.0.OR.TMPI.GT. 219.9) GO TO 50
      CALL PRPSAT(PRSI,TMPI,ZENTH,
```

```
     *  TSO2(16),NO2P(1),NO2T(1),16,159.95,220.05,
     *  O2P1,O2T1,O2H1,PSO2,TSO2,HLO2,HVO2)
      CALL PRPSAT(PRSI,TMPI,ZENTR,
     *  TSO2(16),NO2P(1),NO2T(1),16,159.95,220.05,
     *  O2P1,O2T1,O2S1,PSO2,TSO2,SLO2,SVO2)
      CALL PRPSAT(PRSI,TMPI,ZDENS,
     *  TSO2(16),NO2P(1),NO2T(1),16,159.95,220.05,
     *  O2P1,O2T1,O2D1,PSO2,TSO2,DLO2,DVO2)
      RETURN
C
   22 IF(PRSI.LT.2000.0.OR.PRSI.GT.8000.0) GO TO 50
      CALL ITERP2(PRSI,TMPI,O2P2,O2T2,O2H2,
     *  NO2P(2),NO2T(2),NPX1,NPY1,NO2P(2),ZENTH,N1)
      CALL ITERP2(PRSI,TMPI,O2P2,O2T2,O2S2,
     *  NO2P(2),NO2T(2),NPX1,NPY1,NO2P(2),ZENTR,N1)
      CALL ITERP2(PRSI,TMPI,O2P2,O2T2,O2D2,
     *  NO2P(2),NO2T(2),NPX1,NPY1,NO2P(2),ZDENS,N1)
      RETURN
C
   23 IF(PRSI.LT.2000.0.OR.PRSI.GT.4000.0) GO TO 50
      CALL ITERP2(PRSI,TMPI,O2P3,O2T3,O2H3,
     *  NO2P(3),NO2T(3),NPX1,NPY1,NO2P(3),ZENTH,N1)
      CALL ITERP2(PRSI,TMPI,O2P3,O2T3,O2S3,
     *  NO2P(3),NO2T(3),NPX1,NPY1,NO2P(3),ZENTR,N1)
      CALL ITERP2(PRSI,TMPI,O2P3,O2T3,O2D3,
     *  NO2P(3),NO2T(3),NPX1,NPY1,NO2P(3),ZDENS,N1)
      RETURN
C
C
   30 IF(TMPI.LT.1400.0.OR.TMPI.GT.2000.0) GO TO 50
      IF(PRSI.LT. 100.0.OR.PRSI.GT. 700.0) GO TO 50
      CALL ITERP2(PRSI,TMPI,H2OP1,H2OT1,H2OH1,
     *  NH2OP(1),NH2OT(1),NPX1,NPY1,NH2OP(1),ZENTH,N1)
      CALL ITERP2(PRSI,TMPI,H2OP1,H2OT1,H2OS1,
     *  NH2OP(1),NH2OT(1),NPX1,NPY1,NH2OP(1),ZENTR,N1)
      CALL ITERP2(PRSI,TMPI,H2OP1,H2OT1,H2OD1,
     *  NH2OP(1),NH2OT(1),NPX1,NPY1,NH2OP(1),ZDENS,N1)
      RETURN
C
C
   40 CALL PRPMIX(PRSI,TMPI,OF,CEFF,HMIX,SMIX)
      ZENTH=HMIX
      ZENTR=SMIX
      ZDENS=0.0
      RETURN
C
C
   50 IF(MAT.EQ.1) WRITE(21,51) PRSI,TMPI
      IF(MAT.EQ.2) WRITE(21,52) PRSI,TMPI
      IF(MAT.EQ.3) WRITE(21,53) PRSI,TMPI
      RETURN
C
      END
C**********************************************************************
      SUBROUTINE PRPMIX (P,TMPI,OF,CEFF,HMIX,SMIX)
C
C     PRPMIX - CALCULATES HOT GAS MIXTURE PROPERTIES.
C
      COMMON /H2PRP/
     * H2P1(15),H2T1(11),H2H1(15,11),H2S1(15,11),H2D1(15,11),
```

```
     * H2P2(20),H2T2(11),H2H2(20,11),H2S2(20,11),H2D2(20,11),
     * H2P3(29),H2T3(25),H2H3(29,25),H2S3(29,25),H2D3(29,25),
     * H2P4(23),H2T4(25),H2H4(23,25),H2S4(23,25),H2D4(23,25)
      COMMON /O2PRP/
     * O2P1(13),O2T1(16),O2H1(13,16),O2S1(13,16),O2D1(13,16),
     * O2P2(13),O2T2(17),O2H2(13,17),O2S2(13,17),O2D2(13,17),
     * O2P3(5), O2T3(61),O2H3(5,61), O2S3(5,61), O2D3(5,61)
      COMMON /H2OPRP/
     * H2OP1(7),H2OT1(13),H2OH1(7,13),H2OS1(7,13),H2OD1(7,13)
C
      COMMON /TABLE/
     * NH2P(4),NH2T(4),NO2P(3),NO2T(3),NH2OP(1),NH2OT(1)
      COMMON /STD/
     * HH2REF,HO2REF,HWAREF,SH2REF,SO2REF,SWAREF,SH2A,SO2A,
     * SWAA
C
C
      XMWH2  =     2.0160
      XMWO2  =    31.9988
      XMWH2O =    18.0153
      HCOMB  = -6825.6550
C
      NPX1 = 2
      NPY1 = 2
      ITST1= 0
      ITST2= 0
      ITST3= 0
      ITST4= 0
      ITST5= 0
      ITST6= 0
C
      XF   = 1.0 / (1.0 + OF)
      XO   = 1.0 - XF
      XH2  = XF - XO * 2.0 * CEFF * XMWH2  / XMWO2
      XH2O =      XO * 2.0 * CEFF * XMWH2O / XMWO2
      XO2  = 1.0 - XH2 - XH2O
C
      EH2  = XH2  / XMWH2
      EH2O = XH2O / XMWH2O
      EO2  = XO2  / XMWO2
      ET   = EH2 + EH2O + EO2
C
      YH2  = EH2  / ET
      YH2O = EH2O / ET
      YO2  = 1.0 - YH2 - YH2O
C
      PH2  = P * YH2
      PH2O = P * YH2O
      PO2  = P * YO2
C
      IF(TMPI.LT.1000.0.OR.TMPI.GT.2000.0) ITST1=1
      IF(PH2 .LT.1400.0.OR.PH2 .GT.5800.0) ITST2=1
      CALL ITERP2(PH2,TMPI,H2P4,H2T4,H2H4,
     *  NH2P(4),NH2T(4),NPX1,NPY1,NH2P(4),HH2,N1)
      CALL ITERP2(PH2,TMPI,H2P4,H2T4,H2S4,
     *  NH2P(4),NH2T(4),NPX1,NPY1,NH2P(4),SH2,N1)
C
      IF(TMPI.LT.1400.0.OR.TMPI.GT.2000.0) ITST3=1
      IF(PH2O.LT. 100.0.OR.PH2O.GT. 700.0) ITST4=1
      CALL ITERP2(PH2O,TMPI,H2OP1,H2OT1,H2OH1,
```

```
     *  NH2OP(1),NH2OT(1),NPX1,NPY1,NH2OP(1),HH2O,N1)
      CALL ITERP2(PH2O,TMPI,H2OP1,H2OT1,H2OS1,
     *  NH2OP(1),NH2OT(1),NPX1,NPY1,NH2OP(1),SH2O,N1)
C
      IF(YO2.LT.0.001) THEN
       DHO2 = 0.0
       DSO2 = 0.0
      ELSE
      IF(TMPI.GT. 600.0.AND.TMPI.LT.1500.0) ITST5=1
      IF(PO2 .LT.2000.0.OR. PO2 .GT.4000.0) ITST6=1
      CALL ITERP2(PO2,TMPI,O2P3,O2T3,O2H3,
     *  NO2P(3),NO2T(3),NPX1,NPY1,NO2P(3),HO2,N1)
      CALL ITERP2(PO2,TMPI,O2P3,O2T3,O2S3,
     *  NO2P(3),NO2T(3),NPX1,NPY1,NO2P(3),SO2,N1)
       DHO2 = HO2 - HO2REF
       DSO2 = SO2 - SO2REF + SO2A
      ENDIF
C
   10 DHH2    = HH2 - HH2REF
      DHH2OM = (HH2O - HWAREF) + HCOMB
      DSH2    = SH2 - SH2REF + SH2A
      DSH2O   = SH2O - SWAREF + SWAA
C
      HMIX    = XH2*DHH2 + XH2O*DHH2OM + XO2*DHO2
      SMIX    = XH2*DSH2 + XH2O*DSH2O  + XO2*DSO2
C
C
      IF (ITST1.EQ.1.OR.ITST2.EQ.1) WRITE(21,51) PH2,TMPI
      IF (ITST3.EQ.1.OR.ITST4.EQ.1) WRITE(21,52) PH2O,TMPI
      IF (ITST5.EQ.1.OR.ITST6.EQ.1) WRITE(21,53) PO2,TMPI
C
   51 FORMAT(/3X,'PRPMIX - REQUESTED  PH2 PRS > ',F7.2,2X,
     *  'AND TMP > ',F7.2,2X,'FOR " H2" IS OUT OF RANGE')
   52 FORMAT(/3X,'PRPMIX - REQUESTED PH2O PRS > ',F7.2,2X,
     *  'AND TMP > ',F7.2,2X,'FOR "H2O" IS OUT OF RANGE')
   53 FORMAT(/3X,'PRPMIX - REQUESTED  PO2 PRS > ',F7.2,2X,
     *  'AND TMP > ',F7.2,2X,'FOR " O2" IS OUT OF RANGE')
C
      RETURN
      END
C*********************************************************************
      SUBROUTINE PRPSAT (X,Y,FPROP,TCRT,NX1,NY1,NX2,YL,YH,
     *  PRS1,TMP1,PROP,PRS2,TMP2,PROPL,PROPV)
C
C     PRPSAT - CALCULATES NBS PROPERTIES NEAR SATURATION CURVE
C
      DIMENSION PRS1(1),TMP1(1)
C
      NR1=NX1
      NPX1=2
      NPY1=2
      NPX2=2
C
      ZPLGAS=0.0
      ZPHGAS=0.0
      ZPLLIQ=0.0
      ZPHLIQ=0.0
      ZPROP1=0.0
      ZPROP=0.0
      FPROP=0.0
```

```
      ZTSAT=0.0
      ARGA=0.0
      ARGB=0.0
      ZTSATT=0.0
C
C
      CALL ITERP2(X,Y,PRS1,TMP1,PROP,NX1,NY1,NPX1,NPY1,NR1,ZPROP1,N1)
      FPROP=ZPROP1
      IF(Y.GT.TCRT) GO TO 70
      CALL ITERP1(X,PRS2,TMP2,NX2,NPX2,ZTSAT,N2)
      IF(Y.LT.ZTSAT) GO TO 61
C
C       * *  GAS CALCULATIONS  * *
C
      CALL ITERP1(X,PRS2,PROPV,NX2,NPX2,ZPGAS,N2)
      CALL ITERP2(X,YH,PRS1,TMP1,PROP,NX1,NY1,NPX1,NPY1,NR1,ZTST,N1)
      DTST=ZTST-ZPGAS
      IF(DTST.GT.0.0001) GO TO 50
      ZPLGAS=ZPGAS
      IF(ZPROP1.LT.ZPGAS) GO TO 70
      GO TO 51
   50 ZPHGAS=ZPGAS
      IF(ZPROP1.GT.ZPGAS) GO TO 70
C
   51 LPR=1
   53 PRSD=PRS1(LPR)-0.0001
      IF(PRSD.GT.X) GO TO 52
      LPR=LPR+1
      GO TO 53
C
   52 ARGA=PRS1(LPR)
      CALL ITERP1(ARGA,PRS2,TMP2,NX2,NPX2,ZTSATT,N2)
C
      LTP=1
   54 TMPD=TMP1(LTP)-0.0001
      IF(TMPD.GT.ZTSATT) GO TO 55
      LTP=LTP+1
      GO TO 54
C
   55 ARGB=TMP1(LTP)
      YY=ARGB
      IF(DTST.GT.0.0001) CALL ITERP2(X,YY,PRS1,TMP1,PROP,NX1,NY1,
     *  NPX1,NPY1,NR1,ZPLGAS,N1)
      IF(DTST.LT.0.0001) CALL ITERP2(X,YY,PRS1,TMP1,PROP,NX1,NY1,
     *  NPX1,NPY1,NR1,ZPHGAS,N1)
      ZPROP=ZPHGAS-(ZPHGAS-ZPLGAS)*((ARGB-Y)/(ARGB-ZTSAT))
      FPROP=ZPROP
C
      GO TO 70
C
C       * *  LIQ CALCULATIONS  * *
C
   61 CALL ITERP1(X,PRS2,PROPL,NX2,NPX2,ZPLIQ,N2)
      CALL ITERP2(X,YL,PRS1,TMP1,PROP,NX1,NY1,NPX1,NPY1,NR1,ZTST,N1)
      DTST=ZTST-ZPLIQ
      IF(DTST.GT.0.0001) GO TO 59
      ZPLLIQ=ZPLIQ
      IF(ZPROP1.LT.ZPLIQ) GO TO 70
      GO TO 60
   59 ZPHLIQ=ZPLIQ
```

```
      IF(ZPROP1.GT.ZPLIQ) GO TO 70
C
   60 LPR=1
   63 PRSD=PRS1(LPR)-0.0001
      IF(PRSD.GT.X) GO TO 62
      LPR=LPR+1
      GO TO 63
C
   62 ARGA=PRS1(LPR-1)
      CALL ITERP1(ARGA,PRS2,TMP2,NX2,NPX2,ZTSATT,N2)
C
      LTP=1
   64 TMPD=TMP1(LTP)-0.0001
      IF(TMPD.GT.ZTSATT) GO TO 65
      LTP=LTP+1
      GO TO 64
C
   65 ARGB=TMP1(LTP-1)
      YY=ARGB
      IF(DTST.GT.0.0001) CALL ITERP2(X,YY,PRS1,TMP1,PROP,NX1,NY1,
     *  NPX1,NPY1,NR1,ZPLLIQ,N1)
      IF(DTST.LT.0.0001) CALL ITERP2(X,YY,PRS1,TMP1,PROP,NX1,NY1,
     *  NPX1,NPY1,NR1,ZPHLIQ,N1)
      ZPROP=ZPHLIQ-(ZPHLIQ-ZPLLIQ)*((ZTSAT-Y)/(ZTSAT-ARGB))
      FPROP=ZPROP
C
   70 CONTINUE
C
      RETURN
      END
C**********************************************************************
      SUBROUTINE ITERP1 (X,XT,YT,NX,NPX,Y,NERR)
C
C     ITERP1 - SINGLE INTERPOLATION ROUTINE.
C
      DIMENSION XT(1),YT(1)
      NERR=0
      INTER=1
      NP=NPX
      IF(NX .LT. NP) NP=NX
      IH=NP/2
      I=1
      IF(XT(I)-X)30,20,10
   10 IH=0
   12 NERR=1
      GO TO 70
   13 NERR=2
      GO TO 70
   20 INTER=2
   22 Y=YT(I)
      GO TO 999
   30 I=NX
      IF(XT(I)-X)13,20,40
   40 N1=1
      N2=NX
   45 MP=(N1+N2)/2
   50 IF(XT(MP)-X)52,54,56
   52 N1=MP
      GO TO 60
   54 I=MP
```

```
      GO TO 20
  56  N2=MP
  60  IF((N2-N1) .NE. 1) GO TO 45
C
      IF (N2.GT.(IH+1)) GO TO 65
      I=IH+1
      GO TO 70
  65  I=N2
C
      IF(N2 .GT. I) I=N2
  70  K=I-IH
      N=K+NP-1
      Y=0.
      IF(N-NX)90,90,80
  80  N=NX
      K=NX-NP+1
  90  DO 120 J=K, N
      P=1.0
      DO 110 I=K, N
      IF(I-J)100,110,100
 100  P=P*(X-XT(I))/(XT(J)-XT(I))
 110  CONTINUE
      Y=Y+YT(J)*P
 120  CONTINUE
      GO TO 999
      ENTRY ENTERP (X,XT,YT,Y)
      Y=0.
      GO TO (90,22),INTER
 999  CONTINUE
      RETURN
      END
C*********************************************************************
      SUBROUTINE ITERP2 (X,Y,XT,YT,ZT,NX,NY,NPX,NPY,NR,Z,NERR)
C
C     ITERP2 - DOUBLE INTERPOLATION ROUTINE.
C
      DIMENSION XT(1),YT(1),ZT(NR,1),ZC(15)
      NERRB=0
      NPYY=NPY
      IF(NY .LT. NPY) NPYY=NY
      IH=NPYY/2
      I=1
      IF(YT(I)-Y)30,20,10
   10 IH=0
   12 NERRB=201
      GO TO 70
   13 NERRB=204
      GO TO 70
   20 CALL ITERP1(X,XT,ZT(1,I),NX,NPX,Z,NERRA)
      GO TO 999
   30 I=NY
      IF(YT(I)-Y)13,20,40
   40 N1=1
      N2=NY
   45 MP=(N1+N2)/2
   50 IF(YT(MP)-Y)52,54,56
   52 N1=MP
      GO TO 60
   54 I=MP
      GO TO 20
```

```
 56 N2=MP
 60 IF((N2-N1) .NE. 1) GO TO 45
    I=N2
    IF(I .LT. (IH+1)) I=IH+1
 70 K=I-IH
    N=K+NPYY-1
    IF(N-NY)90,90,80
 80 N=NY
    K=NY-NPYY+1
 90 J=0
    DO 100 I=K, N
    J=J+1
    IF(J .NE. 1) GO TO 95
    CALL ITERP1(X,XT,ZT(1,I),NX,NPX,ZC(J),NERRA)
    GO TO 100
 95 CALL ENTERP(X,XT,ZT(1,I),ZC(J))
100 CONTINUE
    CALL ITERP1(Y,YT(K),ZC,NPYY,NPYY,Z,NERRC)
999 NERR=NERRA+NERRB
    RETURN
    END
```

/N-20-CR

113094

NASA National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No. P. 83
4. Title and Subtitle "Improve SSME Power Balance Model"		5. Report Date September 9, 1992
		6. Performing Organization Code
7. Author(s) Dr. Gerald R. Karr		8. Performing Organization Report No.
		10. Work Unit No.
9. Performing Organization Name and Address The University of Alabama in Huntsville Huntsville, Alabama 35899		11. Contract or Grant No. NAS8-36955/D.O. 115
12. Sponsoring Agency Name and Address NASA Marshall Space Flight Center Marshall Space Flight Center, Alabama 35812		13. Type of Report and Period Covered Final Report
		14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

Effort was dedicated to development and testing of a formal strategy for reconciling uncertain test data with physically limited computational prediction. Specific weaknesses in the logical structure of the current PBM version are described with emphasis given to the main routing subroutines BAL and DATRED. Selected results from a variational analysis of PBM predictions are compared to TTB variational study results to assess PBM predictive capability. The motivation for systematic integration of uncertain test data with computational predictions based on limited physical models is provided. The theoretical foundation for the reconciliation strategy developed in this effort is presented, and results of a reconciliation analysis of the SSME high pressure fuel side turbopump subsystem are examined.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
SSME, TTB	AT01 (1) EM13/L. Smith (1) ONRRR (1) John Butas/EP55 (2 + repro) Vaughan/UAH (1) AP29E (2) NASA/Sci&Tech Info. Facility (1 + repro)

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	83	

NASA FORM 1626 OCT 86